Acta Numerica (1996), *pp.* 1–000

Sequential Quadratic Programming *

Paul T. Boggs
Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, Maryland 20899
USA
Email: boggs@cam.nist.gov

Jon W. Tolle
Departments of Mathematics and Operations Research
University of North Carolina
Chapel Hill, North Carolina 27599
USA
Email: tolle@math.unc.edu

CONTENTS

1. Introduction

Since its popularization in the late 1970s, Sequential Quadratic Programming (SQP) has arguably become the most successful method for solving nonlinearly constrained optimization problems. As with most optimization methods, SQP is not a single algorithm, but rather a conceptual method from which numerous specific algorithms have evolved. Backed by a solid theoretical and computational foundation, both commercial and public domain SQP algorithms have been developed and used to solve a remarkably large set of important practical problems. Recently large scale versions have been devised and tested with promising results.

In this paper we examine the underlying ideas of the SQP method and the theory that establishes it as a framework from which effective algorithms can be derived. In the process we will describe the most popular manifestations of the method, discuss their theoretical properties, and comment on their practical implementations.

The nonlinear programming problem to be solved is

$$\begin{array}{lrcl} \underset{x}{\text{minimize}} & f(\boldsymbol{x}) & & \\ \text{subject to:} & \boldsymbol{h}(\boldsymbol{x}) & = & \mathbf{0} \\ & \boldsymbol{g}(\boldsymbol{x}) & \leq & \mathbf{0} \end{array} \qquad \text{(NLP)}$$

where $f : \mathcal{R}^n \to \mathcal{R}$, $\boldsymbol{h} : \mathcal{R}^n \to \mathcal{R}^m$, and $\boldsymbol{g} : \mathcal{R}^n \to \mathcal{R}^p$. Such problems arise in a variety of applications in science, engineering, industry, and management. In the form (NLP) the problem is quite general; it includes as special cases linear and quadratic programs in which the constraint functions, $\boldsymbol{h}$ and $\boldsymbol{g}$, are affine and f is linear or quadratic. While these problems are important and numerous, the great strength of the SQP method is its ability to solve problems with nonlinear constraints. For this reason it is assumed that (NLP) contains at least one nonlinear constraint function.

The basic idea of SQP is to model (NLP) at a given approximate solution, say $\boldsymbol{x}^k$, by a quadratic programming subproblem, and then to use the solution to this subproblem to construct a better approximation $\boldsymbol{x}^{k+1}$. This process is iterated to create a sequence of approximations that, it is hoped, will converge to a solution $\boldsymbol{x}^*$. Perhaps the key to understanding the performance and theory of SQP is the fact that, with an appropriate choice of quadratic subproblem, the method can be viewed as the natural extension of Newton and quasi-Newton methods to the constrained optimization setting. Thus one would expect SQP methods to share the characteristics of Newton-like methods, namely, rapid convergence when the iterates are close to the solution but possible erratic behavior that needs to be carefully controlled when the iterates are far from a solution. While this correspondence is valid in general, the presence of constraints makes both the analysis and implementation of SQP methods significantly more complex.

Two additional properties of the SQP method should be pointed out. First, SQP is not a feasible-point method; that is, neither the initial point nor any of the subsequent iterates need be *feasible* (a feasible point satisfies all of the constraints of (NLP)). This is a major advantage since finding a feasible point when there are nonlinear constraints may be nearly as hard as solving (NLP) itself. SQP methods can be easily modified so that linear constraints, including simple bounds, are always satisfied. Second, the success of the SQP methods depends on the existence of rapid and accurate algorithms for solving quadratic programs. Fortunately, quadratic programs are easy to solve in the sense that there are a good procedures for their solution. Indeed, when there are only equality constraints the solution to a quadratic program reduces to the solution of a linear system of equations.

When there are inequality constraints a sequence of systems may have to be solved.

A comprehensive theoretical foundation does not guarantee that a proposed algorithm will be effective in practice. In real-life problems hypotheses may not be satisfied, certain constants and bounds may not be computable, matrices may be numerically singular, or the scaling may be poor. A successful algorithm needs adaptive safeguards that deal with these pathologies. The algorithmic details to overcome such difficulties, as well as more mundane questions—how to choose parameters, how to recognize convergence, and how to carry out the numerical linear algebra—are lumped under the term "implementation". While a detailed treatment of this topic is not possible here, we will take care to point out questions of implementation that pertain specifically to the SQP method.

This survey is arranged as follows. In Section 2 we state the basic SQP method along with the assumptions about (NLP) that will hold throughout the paper. We also make some necessary remarks about notation and terminology. Section 3 treats *local convergence,* that is, behavior of the iterates when they are close to the solution. Rates of convergence are provided both in general terms and for some specific SQP algorithms. The goal is not to present the strongest results but to establish the relation between Newton's method and SQP, to delineate the kinds of quadratic models that will yield satisfactory theoretical results, and to place current variations of the SQP method within this scheme.

The term *global* is used in two different contexts in nonlinear optimization and is often the source of confusion. An algorithm is said to be *globally convergent* if, under suitable conditions, it will converge to some local solution from any remote starting point. Nonlinear optimization problems can have multiple local solutions; the *global solution* is that local solution corresponding to the least value of f. SQP methods, like Newton's method and steepest descent, are only guaranteed to find a local solution of (NLP); they should not be confused with algorithms for finding the global solution, which are of an entirely different flavor.

To establish global convergence for constrained optimization algorithms, a way of measuring progress towards a solution is needed. For SQP this is done by constructing a *merit function*, a reduction in which implies that an acceptable step has been taken. In Section 4, two standard merit functions are defined and their advantages and disadvantages for forcing global convergence are considered. In Section 5, the problems of the transition from global to local convergence are discussed. In Sections 4 and 5 the emphasis is on line search methods. Because trust region methods, which have been found to be effective in unconstrained optimization, have been extended to fit into the SQP framework; a brief description of this approach is given in

Section 6. Section 7 is devoted to implementation issues, including those associated with large scale problems.

To avoid interrupting the flow of the presentation, comments on related results and references to the literature are provided at the end of each section. The number of papers on the SQP method and its variants is large and space prohibits us from compiling a complete list of references; we have tried to give enough references to each topic to direct the interested reader to further material.

1.1. Notes and References

The earliest reference to SQP-type algorithms seems to have been in the PhD thesis of Wilson (1963) at Harvard University, in which he proposed the method we call in Section 3 the Newton-SQP algorithm. The development of the secant or variable-metric algorithms for unconstrained optimization in the late 1960's and early 1970's naturally led to the extension of these methods to the constrained problem. The initial work on these methods was done by Mangasarian and his students at the University of Wisconsin. Garcia-Palomares and Mangasarian (1976) investigated an SQP-type algorithm in which the entire Hessian matrix of the Lagrangian, i.e., the matrix of second derivatives with respect to both to x and the multipliers, was updated at each step. Shortly thereafter, Han (Han 1976) and (Han 1977) provided further impetus for the study of SQP methods. In the first paper Han gave local convergence and rate of convergence theorems for the PSB- and BFGS-SQP algorithms for the inequality-constrained problem and, in the second, employed the ℓ_1 merit function to obtain a global convergence theorem in the convex case. In a series of papers presented at conferences, (Powell 1977), (Powell 1978*a*), and (Powell 1978*b*), Han's work was brought to the attention of the general optimization audience. From that time there has been a continuous production of research papers on the SQP method.

As noted, a significant advantage of SQP is that feasible points are not required at any stage of the process. Nevertheless, a version of SQP that always remains feasible has been developed and studied, for example, by Bonnans, Panier, Tits and Zhou (1992).

2. The Basic SQP Method

2.1. Assumptions and Notation

As with virtually all nonlinear problems, it is necessary to make some assumptions on the problem (NLP) that clarify the class of problems for which the algorithm can be shown to perform well. These assumptions, as well as the consequent theory of nonlinear programming, are also needed to describe the SQP algorithm itself. In this presentation we do not attempt to make the weakest assumptions possible, but rather provide what we consider reasonable assumptions under which the main ideas can be illustrated.

We make the blanket assumption that all of the functions in (NLP) are three times continuously differentiable. We denote the gradient of a scalar-valued function by ∇, e.g., $\nabla f(\boldsymbol{x})$. (Here, and throughout the paper, all vectors are assumed to be column vectors; we use the superscript $^{\mathrm{t}}$ to denote transpose.) For vector-valued functions we also use ∇ to denote the Jacobian of the function. For example,

$$\nabla \boldsymbol{h}(\boldsymbol{x}) = \left(\nabla \boldsymbol{h}_1(\boldsymbol{x}), \nabla \boldsymbol{h}_2(\boldsymbol{x}), \ldots, \nabla \boldsymbol{h}_m(\boldsymbol{x})\right).$$

The Hessian of a scalar-valued function, denoted by the letter H, is defined to be the symmetric matrix whose $(i,j)^{th}$ component is

$$Hf(\boldsymbol{x})_{i,j} = \frac{\partial^2 f(\boldsymbol{x})}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j}.$$

Where a function is defined on more than one set of variables, such as the Lagrangian function defined below, the differentiation operators ∇ and H will refer to differentiation *with respect to $\boldsymbol{x}$ only.*

A key function, one that plays a central role in all of the theory of constrained optimization, is the scalar-valued *Lagrangian function* defined by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{x}) + \boldsymbol{u}^{\mathrm{t}} \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}^{\mathrm{t}} \boldsymbol{g}(\boldsymbol{x}) \tag{2.1}$$

where $\boldsymbol{u} \in \mathcal{R}^m$ and $\boldsymbol{v} \in \mathcal{R}^p$ are the *multiplier vectors.* Given a vector $\boldsymbol{x}$, the set of *active* constraints at $\boldsymbol{x}$ consists of the inequality constraints, if any, satisfied as equalities at $\boldsymbol{x}$. We denote the index set of active constraints by

$$\mathcal{I}(\boldsymbol{x}) = \{i \ : \ \boldsymbol{g}_i(\boldsymbol{x}) = 0\}.$$

The matrix $G(\boldsymbol{x})$ made up of the matrix $\nabla \boldsymbol{h}(\boldsymbol{x})$ along with the columns $\nabla \boldsymbol{g}_i(\boldsymbol{x}),\ i \in \mathcal{I}(\boldsymbol{x})$ will be important in describing the basic assumptions and carrying out the subsequent analyses. Assuming that the matrix $G(\boldsymbol{x})$ has full column rank, the null space of $G(\boldsymbol{x})^{\mathrm{t}}$ defines the tangent space to the equality and active inequality constraints at $\boldsymbol{x}$. The projection onto this tangent space can be written as

$$\mathcal{P}(\boldsymbol{x}) = I - G(\boldsymbol{x}) \left(G(\boldsymbol{x})^{\mathrm{t}} G(\boldsymbol{x})\right)^{-1} G(\boldsymbol{x})^{\mathrm{t}}. \tag{2.2}$$

The corresponding projection onto the range space of $G(\boldsymbol{x})$ will be written

$$\mathcal{Q}(\boldsymbol{x}) = I - \mathcal{P}(\boldsymbol{x}). \tag{2.3}$$

For convenience, these projection matrices evaluated at iterates $\boldsymbol{x}^k$ and at a solution $\boldsymbol{x}^*$ will be denoted by $\mathcal{P}^k$, $\mathcal{P}^*$, $\mathcal{Q}^k$, and $\mathcal{Q}^*$. Similarly, we will write

$$H\mathcal{L}^* = H\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$$

throughout the remainder of this paper.

In this paper $\boldsymbol{x}^*$ will represent any particular local solution of (NLP). We assume that the following conditions apply to each such solution.

A1: The *first order necessary conditions* hold, i.e., there exist optimal multiplier vectors $\boldsymbol{u}^*$ and $\boldsymbol{v}^* \geq \mathbf{0}$ such that

$$\nabla\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*) = \nabla f(\boldsymbol{x}^*) + \nabla \boldsymbol{h}(\boldsymbol{x}^*)\boldsymbol{u}^* + \nabla \boldsymbol{g}(\boldsymbol{x}^*)\boldsymbol{v}^* = \mathbf{0}.$$

A2: The columns of $G(\boldsymbol{x}^*)$ are linearly independent.

A3: Strict complementary slackness holds, i.e.,

$$\boldsymbol{g}_i(\boldsymbol{x}^*)\, \boldsymbol{v}_i^* = 0$$

for $i = 1, \ldots, p$ and, if $\boldsymbol{g}_i(\boldsymbol{x}^*) = 0$, then $\boldsymbol{v}_i^* > 0$.

A4: The Hessian of the Lagrangian function with respect to $\boldsymbol{x}$ is positive definite on the null space of $G(\boldsymbol{x}^*)^{\mathrm{t}}$; i.e.,

$$\mathbf{d}^{\mathrm{t}} H\mathcal{L}^* \mathbf{d} > 0$$

for all $\mathbf{d} \neq \mathbf{0}$ such that $G(\boldsymbol{x}^*)^{\mathrm{t}}\mathbf{d} = \mathbf{0}$.

The above conditions, sometimes called the *strong second order sufficient conditions*, in fact guarantee that $\boldsymbol{x}^*$ is an isolated local minimum of (NLP) and that the optimal multiplier vectors $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$ are unique. It should be noted that without strong additional assumptions (NLP) may have multiple local solutions.

We use the term *critical point* to denote a feasible point that satisfies the first order necessary conditions **A1**. A critical point may or may not be a local minimum of (NLP).

In discussing convergence of SQP algorithms the asymptotic *rate of convergence* plays an important role. Three standard measures of convergence rates will be emphasized in this paper. In the definitions that follow, and throughout the paper, the norm $\|\cdot\|$ will refer to the 2-norm unless specifically noted otherwise. Other norms can be used for most of the analysis.

Definition 2.1 Let $\{\boldsymbol{x}^k\}$ be a sequence converging to $\boldsymbol{x}^*$. The sequence is said to converge *linearly* if there exists a positive constant $\xi < 1$ such that

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\| \leq \xi \left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|$$

for all k sufficiently large; *superlinearly* if there exists a sequence of positive constants $\xi_k \to 0$ such that

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\| \leq \xi_k \left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|$$

for all k sufficiently large; and *quadratically* if there exists a positive constant ξ such that

$$\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\| \leq \xi \left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|^2$$

for all k sufficiently large.

These rates, which measure improvement at each step, are sometimes referred to as Q-rates. We will also have occasion to state results in terms of a measure of the average rate of convergence called the R-rate. A sequence $\{\boldsymbol{x}^k\}$ will be said to converge *R-linearly* if the sequence $\{\left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|\}$ is bounded by a sequence that converges Q-linearly to zero. Similar definitions exist for R-superlinear and R-quadratic. There are two relations between R-rate and Q-rate convergence that are of importance for this paper. First, m-step Q-linear convergence (where $k+1$ is replaced by $k+m$ in the above definitions) implies an R-linear rate of convergence and, second, a Q-rate of convergence of a sequence of vectors implies the same R-rate (but not the same Q-rate) of convergence of its components. In the analyses that follow, rates not designated explicitly as Q- or R-rates are assumed to be Q-rates.

2.2. The Quadratic Subproblem

As suggested in the Introduction the SQP method is an iterative method in which, at a current iterate $\boldsymbol{x}^k$, the step to the next iterate is obtained through information generated by solving a quadratic subproblem. The subproblem is assumed to reflect in some way the local properties of the original problem. The major reason for using a quadratic subproblem, i.e., a problem with a quadratic objective function and linear constraints, is that such problems are relatively easy to solve and yet, in their objective function, can reflect the nonlinearities of the original problem. The technical details of solving quadratic programs will not be dealt with here, although the algorithmic issues involved in solving these quadratic subproblems are nontrivial and their resolution will affect the overall performance of the SQP algorithm. Further comments on this are made in Section 7.

A major concern in SQP methods is the choice of appropriate quadratic subproblems. At a current approximation $\boldsymbol{x}^k$ a reasonable choice for the constraints is a linearization of the actual constraints about $\boldsymbol{x}^k$. Thus the quadratic subproblem will have the form

$$\begin{array}{ll} \underset{\boldsymbol{d}_x}{\text{minimize}} & (\mathbf{r}^k)^{\mathrm{t}}\boldsymbol{d}_x + \frac{1}{2}\boldsymbol{d}_x{}^{\mathrm{t}} B_k\, \boldsymbol{d}_x \\ \text{subject to:} & \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{h}(\boldsymbol{x}^k) \;=\; \mathbf{0} \\ & \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{g}(\boldsymbol{x}^k) \;\le\; \mathbf{0} \end{array}$$

where $\boldsymbol{d}_x = \boldsymbol{x} - \boldsymbol{x}^k$. The vector $\mathbf{r}^k$ and the symmetric matrix B_k remain to be chosen.

The most obvious choice for the objective function in this quadratic program is the local quadratic approximation to f at $\boldsymbol{x}^k$. That is, B_k is taken as the Hessian and $\mathbf{r}^k$ as the gradient of f at $\boldsymbol{x}^k$. While this is a reasonable approximation to use if the constraints are linear, the presence of nonlinear constraints makes this choice inappropriate. For example, consider the problem

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & -\boldsymbol{x}_1 - \frac{1}{2}(\boldsymbol{x}_2)^2 \\ \text{subject to:} & (\boldsymbol{x}_1)^2 + (\boldsymbol{x}_2)^2 - 1 = 0. \end{array}$$

The point $(1, 0)$ is a solution satisfying **A1**–**A4**, but at the point $(1+\epsilon, 0)$ the approximating quadratic program is (with $\mathbf{d}$ replacing $\boldsymbol{d}_x$)

$$\begin{array}{ll} \underset{\mathbf{d}}{\text{minimize}} & -\mathbf{d}_1 - \frac{1}{2}(\mathbf{d}_2)^2 \\ \text{subject to:} & \mathbf{d}_1 = -\frac{1}{2}\epsilon(2+\epsilon)(1+\epsilon)^{-1} \end{array}$$

which is unbounded regardless of how small ϵ is. Thus the algorithm that computes $\mathbf{d}$ breaks down for this problem.

To take nonlinearities in the constraints into account while maintaining the linearity of the constraints in the subproblem, the SQP method uses a quadratic model of the Lagrangian function as the objective. This can be justified by noting that conditions **A1**–**A4** imply that $\boldsymbol{x}^*$ is a local minimum for the problem

$$\begin{array}{llcl} \underset{\boldsymbol{x}}{\text{minimize}} & L(\boldsymbol{x}, \boldsymbol{u}^*, \boldsymbol{v}^*) & & \\ \text{subject to:} & \boldsymbol{h}(\boldsymbol{x}) & = & \mathbf{0} \\ & \boldsymbol{g}(\boldsymbol{x}) & \leq & \mathbf{0}. \end{array}$$

Note that the constraint functions are included in the objective function for this equivalent problem. Although the optimal multipliers are not known, approximations $\boldsymbol{u}^k$ and $\boldsymbol{v}^k$ to the multipliers can be maintained as part of the iterative process. Then given a current iterate, $(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)$, the quadratic Taylor series approximation in $\boldsymbol{x}$ for the Lagrangian is

$$\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k) + \nabla\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)^{\mathrm{t}}\boldsymbol{d}_x + \tfrac{1}{2}\boldsymbol{d}_x{}^{\mathrm{t}} H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)\boldsymbol{d}_x.$$

A strong motivation for using this function as the objective function in the quadratic subproblem is that it generates iterates that are identical to those generated by Newton's method when applied to the system composed of the first order necessary condition (condition **A1**) and the constraint equations (including the active inequality constraints). This means that the resulting algorithm will have good local convergence properties. In spite of these local convergence properties there are good reasons to consider choices other than the actual Hessian of the Lagrangian, for example approximating matrices that have properties that permit the quadratic subproblem to be solved at any $\boldsymbol{x}^k$ and the resulting algorithm to be amenable to a global convergence analysis. Letting B_k be an approximation of $H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)$, we can write the quadratic subproblem as:

$$\begin{array}{ll} \underset{\boldsymbol{d_x}}{\text{minimize}} & \nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)^{\mathrm{t}} \boldsymbol{d_x} + \frac{1}{2} \boldsymbol{d_x}^{\mathrm{t}} B_k \, \boldsymbol{d_x} \\ \text{subject to:} & \begin{array}{rcl} \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \boldsymbol{h}(\boldsymbol{x}^k) & = & \mathbf{0} \\ \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \boldsymbol{g}(\boldsymbol{x}^k) & \leq & \mathbf{0}. \end{array} \end{array} \tag{2.4}$$

The form of the quadratic subproblem most often found in the literature, and the one that will be employed here, is

$$\begin{array}{ll} \underset{\boldsymbol{d_x}}{\text{minimize}} & \nabla f(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \frac{1}{2} \boldsymbol{d_x}^{\mathrm{t}} B_k \, \boldsymbol{d_x} \\ \text{subject to:} & \begin{array}{rcl} \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \boldsymbol{h}(\boldsymbol{x}^k) & = & \mathbf{0} \\ \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \boldsymbol{g}(\boldsymbol{x}^k) & \leq & \mathbf{0}. \end{array} \end{array} \tag{QP}$$

These two forms are equivalent for problems with only equality constraints since, by virtue of the linearized constraints, the term $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x}$ is constant and the objective function becomes $\nabla f(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d_x} + \frac{1}{2} \boldsymbol{d_x}^{\mathrm{t}} B_k \boldsymbol{d_x}$. The two subproblems are not quite equivalent in the inequality-constrained case unless the multiplier estimate $\boldsymbol{v}^k$ is zero for all inactive linear constraints. However (QP) is equivalent to (2.4) for the slack-variable formulation of (NLP) given by

$$\begin{array}{ll} \underset{\boldsymbol{x},\, \boldsymbol{z}}{\text{minimize}} & f(\boldsymbol{x}) \\ \text{subject to:} & \begin{array}{rcl} \boldsymbol{h}(\boldsymbol{x}) & = & 0 \\ \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z} & = & \mathbf{0} \\ \boldsymbol{z} & \geq & \mathbf{0} \end{array} \end{array} \tag{2.5}$$

where $\boldsymbol{z} \in \mathcal{R}^p$ is the vector of slack variables. Therefore (QP) can be considered an appropriate quadratic subproblem for (NLP).

The solution $\boldsymbol{d_x}$ of (QP) can be used to generate a new iterate $\boldsymbol{x}^{k+1}$, by taking a step from $\boldsymbol{x}^k$ in the direction of $\boldsymbol{d_x}$. But to continue to the next iteration new estimates for the multipliers are needed. There are several ways in which these can be chosen, but one obvious approach is to use the optimal multipliers of the quadratic subproblem. (Other possibilities will be discussed where appropriate.) Letting the optimal multipliers of (QP) be denoted by $\boldsymbol{u}_{qp}$ and $\boldsymbol{v}_{qp}$, and setting

$$\begin{array}{rcl} \boldsymbol{d_u} & = & \boldsymbol{u}_{qp} - \boldsymbol{u}^k \\ \boldsymbol{d_v} & = & \boldsymbol{v}_{qp} - \boldsymbol{v}^k \end{array}$$

allow the updates of $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v})$ to be written in the compact form

$$\begin{array}{rcl} \boldsymbol{x}^{k+1} & = & \boldsymbol{x}^k + \alpha \boldsymbol{d_x} \\ \boldsymbol{u}^{k+1} & = & \boldsymbol{u}^k + \alpha \boldsymbol{d_u} \\ \boldsymbol{v}^{k+1} & = & \boldsymbol{v}^k + \alpha \boldsymbol{d_v} \end{array} \tag{2.6}$$

for some selection of the *steplength parameter* α. Once the new iterates are constructed, the problem functions and derivatives are evaluated and a prescribed choice of B_{k+1} calculated. It will be seen that the effectiveness of the algorithm is determined in large part by this choice.

2.3. The Basic Algorithm

Since the quadratic subproblem (QP) has to be solved to generate steps in the algorithm, the first priority in analyzing an SQP algorithm is to determine conditions that guarantee that (QP) has a solution. To have a solution the system of constraints of (QP) must have a nonempty feasible set (i.e., the system must be *consistent*) and the quadratic objective function should be bounded below on that set (although a local solution can sometimes exist without this condition). The consistency condition can be guaranteed when $\boldsymbol{x}^k$ is in a neighborhood of $\boldsymbol{x}^*$ by virtue of Assumption **A2** but, depending on the problem, may fail at nonlocal points. Practical means of dealing with this possibility will be considered in Section 7. An appropriate choice of B_k will assure that a consistent quadratic problem will have a solution; the discussion of this point will be a significant part of the analysis of the next two sections.

Assuming that (QP) can be solved, the question of whether the sequence generated by the algorithm will converge must then be resolved. As described in the Introduction, convergence properties generally are classified as either local or global. *Local convergence* results proceed from the assumptions that the initial $\boldsymbol{x}$-iterate is close to a solution $\boldsymbol{x}^*$ and the initial Hessian approximation is, in an appropriate sense, close to $H\mathcal{L}^*$. Conditions on the updating schemes that ensure that the $\boldsymbol{x}^k$ (and the B_k) stay close to $\boldsymbol{x}^*$ (and $H\mathcal{L}^*$) are then formulated. These conditions allow one to conclude that (QP) is a good model of (NLP) at each iteration and hence that the system of equations defined by the first order conditions and the constraints for (QP) are nearly the same as those for (NLP) at $\boldsymbol{x}^*$. Local convergence proofs can be modeled on the proof of convergence of the classical Newton's method, which assumes $\alpha = 1$ in (2.6).

Convergence from a remote starting point is called *global convergence*. As stated in the Introduction, to ensure global convergence the SQP algorithm needs to be equipped with a measure of progress, a merit function ϕ, whose reduction implies progress towards a solution. In order to guarantee that ϕ is reduced at each step a procedure for adjusting the steplength parameter α in (2.6) is required. Using the decrease in ϕ it can be shown that under certain assumptions the iterates will converge to a solution (or, to be precise, a potential solution) even if the initial $\boldsymbol{x}$-iterate, $\boldsymbol{x}^0$, is not close to a solution.

Local convergence theorems are based on Newton's method whereas global convergence theorems are based on descent. Ideally an algorithm should be such that as the iterates get close to the solution, the conditions for local convergence will be satisfied and the local convergence theory will apply without the need for the merit function. In general a global algorithm, although ultimately forcing the $\boldsymbol{x}$-iterates to get close to $\boldsymbol{x}^*$, does not automatically force the other local conditions (such as unit steplengths and close Hessian approximations) to hold and therefore the merit function must be used throughout the iteration process. Since there is no practical way to know when, if ever, the local convergence conditions will hold an implementation's true performance can be deduced only from numerical experience. These issues will be discussed more fully in Section 5.

With this background we can now state a basic SQP algorithm. This template indicates the general steps only, without the numerous details required for a general code.

Basic Algorithm

Given approximations $(\boldsymbol{x}^0, \boldsymbol{u}^0, \boldsymbol{v}^0)$, B_0, and a merit function ϕ, set $k = 0$.

1. Form and solve (QP) to obtain $(\boldsymbol{d}_x, \boldsymbol{d}_u, \boldsymbol{d}_v)$.
2. Choose steplength α so that

$$\phi(\boldsymbol{x}^k + \alpha \boldsymbol{d}_x) < \phi(\boldsymbol{x}^k).$$

3. Set

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \alpha \boldsymbol{d}_x \\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \alpha \boldsymbol{d}_u \\ \boldsymbol{v}^{k+1} &= \boldsymbol{v}^k + \alpha \boldsymbol{d}_v. \end{aligned}$$

4. Stop if converged.
5. Compute B_{k+1}.
6. Set $k := k + 1$; go to 1.

2.4. Notes and References

The basic second order sufficient conditions as well as a description of the major theoretical ideas of finite-dimensional nonlinear constrained optimization can be found in numerous sources. See, e.g., (Luenberger 1984), (Nash and Sofer 1995), or (Gill, Murray and Wright 1981). The basic definitions for various rates of convergence and the relations among them can be found in (Ortega and Rheinboldt 1970) and (Dennis and Schnabel 1983).

The trust region methods discussed in Section 6 use different quadratic subproblems than those given here. Also Fukushima (1986) considers a different quadratic subproblem in order to avoid the Maratos effect discussed in Section 5.

3. Local Convergence

In this section the theory of local convergence for the most common versions of the SQP method is developed. The local convergence analysis establishes conditions under which the iterates converge to a solution and at what rate, given that that the starting data (e.g., x^0, u^0, v^0, B_0) are sufficiently close to the corresponding data at a solution, x^*. As will be seen, it is the method of generating the matrices B_k that will be the determining factor in obtaining the convergence results.

An SQP method can have excellent theoretical local convergence properties; quadratic, superlinear, or two-step superlinear convergence of the x - iterates can be achieved by requiring the B_k to approximate $H\mathcal{L}^*$ in an appropriate manner. Although each of these local convergence rates is achievable in practice under certain conditions, the need for a globally convergent algorithm further restricts the choice of the B_k. Certain choices of the B_k have led to algorithms that extensive numerical experience has shown to be quite acceptable. Therefore, while there is a gap between what is theoretically possible and what has been achieved in terms of local convergence, this discrepancy should not obscure the very real progress that the SQP methods represent in solving nonlinearly constrained optimization problems.

Two important assumptions simplify the presentation. First, it will be assumed that the active inequality constraints for (NLP) at $x^*$are known. As will be discussed in Section 5, this assumption can be justified for many of the SQP algorithms because the problem (QP) at x^k will have the same active constraints as (NLP) at x^* when x^k is near x^*. The fact that the active constraints, and hence the inactive constraints, are correctly identified at x^k means that those inequality constraints that are inactive for (NLP) at x^* can be ignored and those that are active can be changed to equality constraints without changing the solution of (QP). Thus, under this assumption, only equality-constrained problems need be considered for the local analysis. For the remainder of this section (NLP) will be assumed to be an equality-constrained problem and the reference to inequality multiplier vectors (the $v^{\mathrm{t}} g(x)$ term) will be eliminated from the Lagrangian. For reference, we rewrite the quadratic subproblem with only equality constraints:

$$\begin{array}{ll} \underset{d_x}{\text{minimize}} & \nabla f(x^k)^{\mathrm{t}} d_x + \frac{1}{2} d_x{}^{\mathrm{t}} B_k\, d_x \\ \text{subject to:} & \nabla h(x^k)^{\mathrm{t}} d_x + h(x^k) \;=\; \mathbf{0}. \end{array} \qquad (ECQP)$$

The second assumption follows from the fact that the local convergence for SQP methods, as for many optimization methods, is based on Newton's method; that is, the convergence and rate of convergence results are obtained by demonstrating an asymptotic relation between the iterates of the method being analyzed and the classical Newton steps. Because Newton's method for a system with a nonsingular Jacobian at the solution requires that unit steps be taken to obtain rapid convergence, we assume that the merit function allows $\alpha = 1$ to be used in the update formulas (2.6). Similar results to those given in this section can be obtained if the steplengths converge to one sufficiently quickly.

For the equality-constrained problem a good initial estimate of $\boldsymbol{x}$ can be used to obtain a good estimate for the optimal multiplier vector. The first order necessary condition, **A1**, together with condition **A2**, leads to the formula

$$\boldsymbol{u}^* = -[\nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} A \, \nabla \boldsymbol{h}(\boldsymbol{x}^*)]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} A \, \nabla f(\boldsymbol{x}^*). \tag{3.1}$$

for any nonsingular matrix A that is positive definite on the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}}$. In particular, if A is taken to be the identity matrix, then (3.1) defines the least squares solution of the first order necessary conditions. By our smoothness assumptions, it follows that

$$\boldsymbol{u}^0 = -[\nabla \boldsymbol{h}(\boldsymbol{x}^0)^{\mathrm{t}} \nabla \boldsymbol{h}(\boldsymbol{x}^0)]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^0)^{\mathrm{t}} \nabla f(\boldsymbol{x}^0) \tag{3.2}$$

can be made arbitrarily close to $\boldsymbol{u}^*$ by choosing $\boldsymbol{x}^0$ close to $\boldsymbol{x}^*$. Consequently, no additional assumption will be made about an initial optimal multiplier estimate for the local convergence analysis.

Denoting the optimal multiplier for (ECQP) by $\boldsymbol{u}_{qp}$ we see that the first order conditions for this problem are

$$\begin{aligned} B_k \, \boldsymbol{d}_x + \nabla \boldsymbol{h}(\boldsymbol{x}^k) \boldsymbol{u}_{qp} &= -\nabla f(\boldsymbol{x}^k) \\ \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x &= -\boldsymbol{h}(\boldsymbol{x}^k). \end{aligned}$$

If, as discussed in Section 2, we set

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}_{qp} = \boldsymbol{u}^k + \boldsymbol{d}_u \tag{3.3}$$

then the above equations become

$$B_k \, \boldsymbol{d}_x + \nabla \boldsymbol{h}(\boldsymbol{x}^k) \boldsymbol{d}_u = -\nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k) \tag{3.4}$$

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x = -\boldsymbol{h}(\boldsymbol{x}^k). \tag{3.5}$$

We are now in a position to begin the analysis of the SQP methods.

3.1. The Newton SQP Method

The straightforward SQP method derived from setting

$$B_k = H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)$$

will be analyzed first. Assuming that $\boldsymbol{x}^0$ is close to $\boldsymbol{x}^*$ it follows from (3.2) that $\boldsymbol{u}^0$ can be presumed to be close to $\boldsymbol{u}^*$ and hence that $H\mathcal{L}(\boldsymbol{x}^0, \boldsymbol{u}^0)$ is close to $H\mathcal{L}^*$. The local convergence for the SQP algorithm now follows from the application of Newton's method to the nonlinear system of equations obtained from the first order necessary conditions and the constraint equation:

$$\Psi(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \nabla\mathcal{L}(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{h}(\boldsymbol{x}) \end{bmatrix} = \mathbf{0}.$$

From assumptions **A1** and **A4** the Jacobian of this system at the solution,

$$J(\boldsymbol{x}^*,\boldsymbol{u}^*) = \begin{bmatrix} H\mathcal{L}^* & \nabla\boldsymbol{h}(\boldsymbol{x}^*) \\ \nabla\boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} & 0 \end{bmatrix}, \tag{3.6}$$

is nonsingular. Therefore, the Newton iteration scheme

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \mathrm{s}_x \\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \mathrm{s}_u \end{aligned}$$

where the vector $\mathrm{s} = (\mathrm{s}_x, \mathrm{s}_u)$ is the solution to

$$J(\boldsymbol{x}^k,\boldsymbol{u}^k)\,\mathrm{s} = -\Psi(\boldsymbol{x}^k,\boldsymbol{u}^k), \tag{3.7}$$

yields iterates that converge to $(\boldsymbol{x}^*,\boldsymbol{u}^*)$ quadratically provided $(\boldsymbol{x}^0,\boldsymbol{u}^0)$ is sufficiently close to $(\boldsymbol{x}^*,\boldsymbol{u}^*)$. The equations (3.7) are identical to equations (3.4) and (3.5) with $B_k = H\mathcal{L}(\boldsymbol{x}^k,\boldsymbol{u}^k)$, $\boldsymbol{d}_x = \mathrm{s}_x$, and $\boldsymbol{d}_u = \mathrm{s}_u$. Consequently the iterates $(\boldsymbol{x}^{k+1},\boldsymbol{u}^{k+1})$ are exactly those generated by the SQP algorithm. For this reason we call this version of the algorithm the (local) Newton SQP method. The basic local convergence results are summarized in the following theorem:

Theorem 3.1 *Let $\boldsymbol{x}^0$ be an initial estimate of the solution to (NLP) and let $\boldsymbol{u}^0$ be given by (3.2). Suppose that the sequence of iterates $\{(\boldsymbol{x}^k,\boldsymbol{u}^k)\}$ is generated by*

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \boldsymbol{d}_x \\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \boldsymbol{d}_u \end{aligned}$$

where $\boldsymbol{d}_x$ and $\boldsymbol{u}_{qp} = \boldsymbol{u}^k + \boldsymbol{d}_u$ are the solution and multiplier of the quadratic program (ECQP) with $B_k = H\mathcal{L}(\boldsymbol{x}^k,\boldsymbol{u}^k)$. Then, if $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|$ is sufficiently small, the sequence of iterates in $(\boldsymbol{x},\boldsymbol{u})$-space is well-defined and converges quadratically to the pair $(\boldsymbol{x}^*,\boldsymbol{u}^*)$.

It is important to emphasize that this version of the SQP algorithm always requires a unit step in the variables; there is no line search to try to determine a better point. If a line search is used (the so-called damped Newton method) then the rate of convergence of the iterates can be significantly decreased—usually to linear.

In a sense, the Newton version of the SQP algorithm can be thought of as the ideal algorithm for solving the nonlinear program: it provides rapid convergence with no requirement for line searches and no necessity for the introduction of additional parameters. Of course, in this form it has little practical value. It is often difficult to choose an initial point close enough to the true solution to guarantee that the Newton algorithm will converge to a minimum of f. When remote from the solution $H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)$ cannot be assumed to be positive definite on the appropriate subspace and hence a solution to the subproblem (ECQP) cannot be guaranteed to exist. The value of this Newton SQP method is that it provides a standard against which the local behavior of other versions can be measured. In fact, to obtain superlinear convergence it is necessary and sufficient that the steps aproach the Newton steps as the solution is neared. Despite the disadvantages, the use of $H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)$ often makes (ECQP) an excellent model of (NLP) and its use, combined with some of the techniques of Sections 6 and 7, can lead to effective algorithms. The actual computation of $H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)$ can be accomplished efficiently by using finite difference techniques or by automatic differentiation methods.

3.2. Conditions for Local Convergence

Here we discuss the theoretical properties of the approximation matrices, B_k, that are sufficient to guarantee that the local SQP method will give Newton-like convergence results. In the following subsections the practical attempts to generate matrices that satisfy these properties will be described.

In order to formulate conditions on B_k that will yield locally convergent SQP algorithms, the following assumption on $H\mathcal{L}^*$ will be imposed.

A5: The matrix $H\mathcal{L}^*$ is nonsingular.

In light of **A4** most of the results that follow could be extended to cover the case where this condition is not satisfied, but the resulting complexity of the arguments would clutter the exposition.

The following conditions on the matrix approximations will be referenced in the remainder of the paper.

B1: The matrices B_k are uniformly positive definite on the null spaces of the matrices $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}$, i.e.,there exists a $\beta_1 > 0$ such that for each k

$$\mathbf{d}^{\mathrm{t}} B_k \mathbf{d} \geq \beta_1 \|\mathbf{d}\|^2$$

for all $\mathbf{d}$ satisfying

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \mathbf{d} = \mathbf{0}.$$

B2: The sequence $\{B_k\}$ is uniformly bounded, i.e., there exists a $\beta_2 > 0$ such that for each k

$$\|B_k\| \leq \beta_2.$$

B3: The matrices B_k have uniformly bounded inverses, i.e., there exists a

$\beta_3 > 0$ such that for each k, ${B_k}^{-1}$ exists and

$$\left\| {B_k}^{-1} \right\| \leq \beta_3.$$

As the SQP methods require the solution of the quadratic subproblem at each step, it is imperative that the choice of matrices B_k make that possible. The second order sufficient conditions for a solution to (ECQP) to exist require that the matrix B_k be positive definite on the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}$ (cf. **A4**). Condition **B1** is a slight strengthening of this requirement. Assumption **A5** and the fact that the matrices B_k are to approximate the Hessian of the Lagrangian suggest that conditions **B2** and **B3** are not unreasonable.

Under assumptions **B1**–**B3** (3.4) and (3.5) have a solution provided $(\boldsymbol{x}^k, \boldsymbol{u}^k)$ is sufficiently close to $(\boldsymbol{x}^*, \boldsymbol{u}^*)$. Moreover, the solutions can be written in the form

$$\boldsymbol{d}_u = [\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} {B_k}^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^k)]^{-1} [\boldsymbol{h}(\boldsymbol{x}^k) - \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} {B_k}^{-1} \nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)] \quad (3.8)$$

and

$$\boldsymbol{d}_x = -{B_k}^{-1} \nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^{k+1}) \quad (3.9)$$

where $\boldsymbol{u}^{k+1}$ is given by (3.3). In particular (3.8) leads to the relation

$$\begin{aligned} \boldsymbol{u}^{k+1} &= [\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} {B_k}^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^k)]^{-1} [\boldsymbol{h}(\boldsymbol{x}^k) - \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} {B_k}^{-1} \nabla f(\boldsymbol{x}^k)] \\ &\equiv W(\boldsymbol{x}^k, B_k). \end{aligned}$$

Setting $A = {B_k}^{-1}$ in (3.1) yields

$$\boldsymbol{u}^* = W(\boldsymbol{x}^*, B_k).$$

It can now be deduced that

$$\begin{aligned} \boldsymbol{u}^{k+1} - \boldsymbol{u}^* &= W(\boldsymbol{x}^k, B_k) - W(\boldsymbol{x}^*, B_k) \\ &= [\nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} {B_k}^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^*)]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} {B_k}^{-1} (B_k - H\mathcal{L}^*)(\boldsymbol{x}^k - \boldsymbol{x}^*) \\ &\quad + \boldsymbol{w}_k \end{aligned} \quad (3.10)$$

where, by assumptions **B2** and **B3**,

$$\boldsymbol{w}_k \leq \kappa \left\| \boldsymbol{x}^k - \boldsymbol{x}^* \right\|^2$$

for some κ independent of k. It should be noted that (3.10) can be used in conjunction with Theorem 3.1 to prove that the sequence $\{\boldsymbol{x}^k\}$ is quadratically convergent when the Newton SQP method is used and $H\mathcal{L}^*$ is nonsingular. In that case $\{\boldsymbol{u}^k\}$ converges R-quadratically.

Equations (3.9) and (3.10) and **A2** now yield

$$\begin{aligned} \boldsymbol{x}^{k+1} - \boldsymbol{x}^* &= \boldsymbol{x}^k - \boldsymbol{x}^* - {B_k}^{-1} \left[\nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^{k+1}) - \nabla \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}^*) \right] \\ &= {B_k}^{-1} \left[(B_k - H\mathcal{L}^*)(\boldsymbol{x}^k - \boldsymbol{x}^*) - \nabla \boldsymbol{h}(\boldsymbol{x}^*)(\boldsymbol{u}^{k+1} - \boldsymbol{u}^*) \right] \end{aligned}$$

$$
\begin{aligned}
& +\mathcal{O}\left(\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2\right) \\
= \quad & B_k{}^{-1}V_k(B_k-H\mathcal{L}^*)(\boldsymbol{x}^k-\boldsymbol{x}^*)+\mathcal{O}\left(\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2\right) \qquad (3.11)
\end{aligned}
$$

where

$$V_k = I - \nabla \boldsymbol{h}(\boldsymbol{x}^*)[\nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} B_k{}^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^*)]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} B_k{}^{-1}.$$

The projection operator defined by (2.2), which in the equality-constrained case has the form

$$\mathcal{P}(\boldsymbol{x}) = I - \nabla \boldsymbol{h}(\boldsymbol{x})[\nabla \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}} \nabla \boldsymbol{h}(\boldsymbol{x})]^{-1} \nabla \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}},$$

satisfies

$$V_k \mathcal{P}^k = V_k.$$

Thus (3.11) leads to the inequality

$$
\begin{aligned}
\left\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\right\| \quad \leq \quad & \left\|B_k{}^{-1}\right\| \|V_k\| \left\|\mathcal{P}^k(B_k-H\mathcal{L}^*)(\boldsymbol{x}^k-\boldsymbol{x}^*)\right\| \\
& +\mathcal{O}\left(\left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\|^2\right). \qquad (3.12)
\end{aligned}
$$

Using induction the above analysis can be made rigorous to yield the following local convergence theorem.

Theorem 3.2 *Let assumptions* **B1–B3** *hold. Then there exist positive constants* ϵ *and* γ *such that if*

$$\left\|\boldsymbol{x}^0-\boldsymbol{x}^*\right\| < \epsilon,$$

$$\left\|\boldsymbol{u}^0-\boldsymbol{u}^*\right\| < \epsilon,$$

and

$$\left\|\mathcal{P}^k(B_k-H\mathcal{L}^*)(\boldsymbol{x}^k-\boldsymbol{x}^*)\right\| < \gamma \left\|\boldsymbol{x}^k-\boldsymbol{x}^*\right\| \qquad (3.13)$$

for all k, *then the sequences* $\{\boldsymbol{x}^k\}$ *and* $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ *generated by the SQP algorithm are well defined and converge linearly to* $\boldsymbol{x}^*$ *and* $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ *respectively. The sequence* $\{\boldsymbol{u}^k\}$ *converges R-linearly to* $\boldsymbol{u}^*$.

Condition (3.13) is, in conjuction with the other hypotheses of the theorem, almost necessary for linear convergence. In fact, if the other hypotheses of the theorem hold and the sequence $\{\boldsymbol{x}^k\}$ converges linearly, then there exists a ξ such that

$$\left\|\mathcal{P}^k(B_k-H\mathcal{L}^*)(\boldsymbol{x}^k-\boldsymbol{x}^*)\right\| < \xi \left\|(\boldsymbol{x}^k-\boldsymbol{x}^*)\right\|$$

for all k. These results and those that follow indicate the crucial role that the projection of $(B_k - H\mathcal{L}^*)$ plays in the theory. The inequality (3.13) is guaranteed by either of the stronger conditions:

$$\left\| \mathcal{P}^k (B_k - H\mathcal{L}^*) \right\| \leq \gamma$$

or

$$\|(B_k - H\mathcal{L}^*)\| \leq \gamma \tag{3.14}$$

which are easier to verify in practice.

In order to satisfy (3.14) it is not necessary that the approximations converge to the true Hessian but only that the growth of the difference $\|B_k - H\mathcal{L}^*\|$ be kept under some control. In the quasi-Newton theory for unconstrained optimization this can be accomplished if the approximating matrices have a property called *bounded deterioration.* This concept can be generalized from unconstrained optimization to the constrained setting in a straightforward way.

Definition 3.1 A sequence of matrix approximations, $\{B_k\}$, for the SQP method is said to have the *bounded deterioration* property if there exist constants α_1 and α_2 independent of k such that

$$\|B_{k+1} - H\mathcal{L}^*\| \leq (1 + \alpha_1 \sigma_k) \|B_k - H\mathcal{L}^*\| + \alpha_2 \sigma_k \tag{3.15}$$

where

$$\sigma_k = \max\{\left\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\right\|, \left\|\boldsymbol{x}^k - \boldsymbol{x}^*\right\|, \left\|\boldsymbol{u}^{k+1} - \boldsymbol{u}^*\right\|, \left\|\boldsymbol{u}^k - \boldsymbol{u}^*\right\|\}.$$

It seems plausible that the bounded deterioration condition when applied to the SQP process will lead to a sequence of matrices that satisfy **B2**, **B3**, and (3.13) provided the initial matrix is close to the Hessian of the Lagrangian and $(\boldsymbol{x}^k,\boldsymbol{u}^k)$ is close to $(\boldsymbol{x}^*,\boldsymbol{u}^*)$. Indeed this is the case, as can be shown by induction to lead to the following result.

Theorem 3.3 *Suppose that the sequence of iterates $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ is generated by the SQP algorithm and the sequence $\{B_k\}$ of symmetric matrix approximations satisfies* **B1** *and (3.15). If $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|$ and $\|B_0 - H\mathcal{L}^*\|$ are sufficiently small and $\boldsymbol{u}^0$ is given by* (3.2) *then the hypotheses of Theorem 3.2 hold.*

The linear convergence for the iterates guaranteed by the above theorem is hardly satisfactory in light of the quadratic convergence of the Newton SQP method. As would be expected, a stronger relation between the approximation matrices, B_k, and the Hessian of the Lagrangian is needed to improve the rate of convergence; however, this relation still depends only on the projection of the difference between the approximation and the true Hessian. With some effort the following theorem can be deduced as a consequence of the inequalities above.

Theorem 3.4 *Let assumptions* **B1**–**B3** *hold and let the sequence* $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$

be generated by the SQP algorithm. Assume that $\boldsymbol{x}^k \to \boldsymbol{x}^*$. *Then the sequence* $\{\boldsymbol{x}^k\}$ *converges to* $\boldsymbol{x}^*$ *superlinearly if and only if the matrix approximations satisfy*

$$\lim_{k\to\infty} \frac{\left\| \mathcal{P}^k(B_k - H\mathcal{L}^*)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \right\|}{\left\| (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \right\|} = 0. \tag{3.16}$$

If this equation holds then the sequence $\{\boldsymbol{u}^k\}$ *converges R-superlinearly to* $\boldsymbol{u}^*$ *and the sequence* $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ *converges superlinearly.*

Not that this theorem requires convergence of the $\boldsymbol{x}$-iterates; (3.16) does not appear to be enough to yield convergence by itself. A slightly different result that uses a two-sided approximation of the Hessian approximation can be obtained by writing (3.16) as

$$\lim_{k\to\infty} \left\{ \frac{\mathcal{P}^k(B_k - H\mathcal{L}^*)\mathcal{P}^k\, s_k}{\|s_k\|} + \frac{\mathcal{P}^k(B_k - H\mathcal{L}^*)\mathcal{Q}^k\, s_k}{\|s_k\|} \right\} = 0 \tag{3.17}$$

where $s_k = (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$. It can be shown that if only the first term goes to zero then a weaker form of superlinear convergence holds.

Theorem 3.5 *Let assumptions* **B1**–**B3** *hold and let the sequence* $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ *be generated by the SQP algorithm. Assume that* $\boldsymbol{x}^k \to \boldsymbol{x}^*$. *Then, if*

$$\lim_{k\to\infty} \frac{\left\| \mathcal{P}^k(B_k - H\mathcal{L}^*)\mathcal{P}^k(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \right\|}{\left\| (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \right\|} = 0, \tag{3.18}$$

the sequence $\{\boldsymbol{x}^k\}$ *converges to* $\boldsymbol{x}^*$ *two-step superlinearly.*

If the sequence $\{\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\}$ approaches zero in a manner tangent to the null space of the Jacobians, i.e.,

$$\lim_{k\to\infty} \frac{\mathcal{Q}^k(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)}{\left\| \boldsymbol{x}^{k+1} - \boldsymbol{x}^k \right\|} = 0,$$

then (3.18) implies (3.16) and superlinear convergence results. This *tangential convergence* has been observed in practice for some of the methods discussed below.

It has been difficult to find useful updating schemes for the B_k that satisfy these conditions for linear and superlinear convergence. Two basic approaches have been tried for generating good matrix approximations. In *full* Hessian approximations the matrices B_k are chosen to approximate $H\mathcal{L}^*$ while in *reduced* Hessian approximations only matrices that approximate the Hessian on the null space of the Jacobians of the constraints are computed. Each of these methods will be described in turn.

3.3. Full Hessian Approximations

An obvious choice of the matrix B_k is a finite difference approximation of the Hessian of the Lagrangian at (x^k, u^k). It is clear from the Theorems 3.2 and 3.4 that if a finite difference method is used, the resulting sequence will be superlinearly convergent if the finite difference step size goes to zero and will have rapid linear convergence for fixed step size. Of course using finite difference approximations, while not requiring evaluation of second derivatives, suffers from the same global difficulties as the Newton SQP method described in the earlier subsection.

As has been seen, one way of obtaining convergence is to use a sequence of Hessian approximations that satisfies the bounded deterioration property. However, this property is not, by itself, enough to guarantee that (3.16) is satisfied and hence it does not guarantee superlinear convergence. The condition (3.16) essentially requires the component of the step generated by the Hessian approximation in the direction of the null space to converge asymptotically to the corresponding component of the step generated by the Newton SQP method. In the following a class of approximations called *secant* approximations, which satisfy a version of this condition and have the bounded deterioration property as well, is considered.

Under the smoothness assumptions of this paper the Lagrangian satisfies

$$\nabla\mathcal{L}(x^{k+1}, u^{k+1}) - \nabla\mathcal{L}(x^k, u^{k+1}) \approx H\mathcal{L}(x^{k+1}, u^{k+1})(x^{k+1} - x^k)$$

with equality if the Lagrangian is quadratic in x. As a result, it makes sense to approximate the Hessian of the Lagrangian at (x^{k+1}, u^{k+1}) by requiring B_{k+1} to satisfy

$$B_{k+1}(x^{k+1} - x^k) = \nabla\mathcal{L}(x^{k+1}, u^{k+1}) - \nabla\mathcal{L}(x^k, u^{k+1}), \tag{3.19}$$

especially since this approximation strongly suggests that (3.16) is likely to be satisfied near the solution. Equation (3.19) is called the *secant* equation; it plays an important role in the algorithmic theory of nonlinear systems and unconstrained optimization.

A common procedure for generating the Hessian approximations that satisfy (3.19) is to compute (x^{k+1}, u^{k+1}) with a given B_k and then to update B_k according to

$$B_{k+1} = B_k + U_k$$

where U_k is a rank-one or rank-two matrix that depends on the values of B_k, x^k, u^k, x^{k+1}, and u^{k+1}. In choosing secant approximations that have the bounded deterioration property, it is natural to look to those updating schemes of this type that have been developed for unconstrained optimization.

The rank-two Powell-symmetric-Broyden (PSB) formula gives one such updating scheme. For the constrained case this update is given by

$$\begin{aligned} B_{k+1} &= B_k + \frac{1}{(s^t s)}[(y - B_k s)s^t + s(y - B_k s)^t] \\ &\quad - \frac{(y - B_k s)^t s}{(s^t s)^2} s\, s^t, \end{aligned} \tag{3.20}$$

where

$$s = x^{k+1} - x^k \tag{3.21}$$

and

$$y = \nabla \mathcal{L}(x^{k+1}, u^{k+1}) - \nabla \mathcal{L}(x^k, u^{k+1}). \tag{3.22}$$

The PSB-SQP algorithm which employs this update has been shown to have the desired local convergence properties.

Theorem 3.6 *Suppose that the sequence of iterates $\{(x^k, u^k)\}$ is generated by the SQP algorithm using the sequence $\{B_k\}$ of matrix approximations generated by the PSB update formulas, (3.20)–(3.22). Then, if $\|x^0 - x^*\|$ and $\|B_0 - H\mathcal{L}(x^*, u^*)\|$ are sufficiently small and u^0 is given by (3.2), the sequence $\{B_k\}$ is of bounded deterioration and the iterates (x^k, u^k) are well defined and converge superlinearly to (x^*, u^*). In addition the x-iterates converge superlinearly and the multipliers converge R-superlinearly.*

Note that there is no assumption of positive definiteness here. In fact, while **B2** and **B3** are satisfied as a result of the bounded deterioration, **B1** does not necessarily hold. Consequently, d_x and d_u are not necessarily solutions of (ECQP), but of the first order conditions (3.4) and (3.5).

As a practical method, the PSB-SQP algorithm has the advantage over the Newton SQP method of not requiring the computation of the Hessian of the Lagrangian (but, as a result, yielding only superlinear rather than quadratic convergence). However, because the matrices are not necessarily positive definite it suffers from the same serious drawback; the problem (ECQP) may not have a solution if the initial starting point is not close to the solution. Consequently, it does not appear to be useful in establishing a globally convergent algorithm (see, however, Section 6). As mentioned above, solving (ECQP) requires that the matrices be positive definite on the appropriate subspace for each subproblem. One way to enforce this is to require the matrices to be positive definite.

A rank-two updating scheme that is considered to be the most effective for unconstrained problems and has useful positive definite properties is the BFGS method. The formula for this update, generalized to be applicable to the constrained problem, is

$$B_{k+1} = B_k + \frac{\mathrm{y}\,\mathrm{y}^t}{\mathrm{s}^t \mathrm{y}} - \frac{B_k \mathrm{s}\,\mathrm{s}^t B_k}{\mathrm{s}^t B_k \mathrm{s}}, \tag{3.23}$$

where s and y are as in (3.21) and (3.22). The important qualities of the matrices generated by this formula are that they have the bounded deterioration property and satisfy a hereditary positive definiteness condition. The latter states that if B_k is positive definite and

$$\mathrm{y}^{\mathrm{t}}\mathrm{s} > 0 \tag{3.24}$$

then B_{k+1} is positive definite. If the matrix approximations are positive definite then (ECQP) is easily solved and the SQP steps are well defined. Unfortunately, because the Hessian of the Lagrangian at (x^*, u^*) need only be positive definite on a subspace, it need not be the case that (3.24) is satisfied and hence the algorithm may break down, even close to the solution. However, if the Hessian of the Lagrangian is positive definite, (3.24) is valid provided B_k is positive definite and (x^k, u^k) is close to (x^*, u^*). In this case this BFGS-SQP algorithm has the same local convergence properties as the PSB-SQP method.

Theorem 3.7 *Suppose that $H\mathcal{L}^*$ is positive definite and let B_0 be an initial positive definite matrix. Suppose that the sequence of iterates $\{(x^k, u^k)\}$ is generated by the SQP algorithm using the sequence of matrix approximations generated by the BFGS update (3.23). Then, if $\|x^0 - x^*\|$ and $\|B_0 - H\mathcal{L}(x^*, u^*)\|$ are sufficiently small and u^0 is given by* (3.2), *the sequence $\{B_k\}$ is of bounded deterioration and (3.16) is satisfied. Therefore, the iterates (x^k, u^k) converge superlinearly to the pair (x^*, u^*). In addition the x-iterates converge superlinearly and the multipliers converge R-superlinearly.*

The assumption that $H\mathcal{L}^*$ is positive definite allows B_0 to be both positive definite and close to $H\mathcal{L}^*$. The requirement that $H\mathcal{L}^*$ be positive definite is satisfied, for example, if (NLP) is convex, but cannot be assumed in general. However, because the positive definiteness of B_k permits the solution of the quadratic subproblems independent of the nearness of the iterate to the solution, the BFGS method has been the focus of vigorous research efforts to adapt it to the general case. Several of these efforts have resulted in implementable algorithms.

One scheme, which we call the Powell-SQP method, is to maintain the positive definite property of the Hessian approximations by modifying the update formula (3.23) by replacing y in (3.22) by

$$\hat{y} = \theta\, y + (1 - \theta)\, B_k\, s \tag{3.25}$$

for some $\theta \in (0, 1]$. With this modification, the condition (3.24) can always be satisfied although the updates no longer satisfy the secant condition. Nevertheless, it has been shown that a specific choice of θ leads to a sequence, $\{x^k\}$, that converges R-superlinearly to x^* *provided that the sequence converges.* Unfortunately, no proof of local covergence has been found although algorithms based on this procedure have proven to be quite successful in practice.

A second approach is to transform the problem so that $H\mathcal{L}^*$ is positive definite and then to apply the BFGS-SQP method on the transformed problem. It is a common trick in developing algorithms for solving equality constrained optimization problems to replace the objective function in (NLP) by the function

$$f_A(\boldsymbol{x}) = f(x) + \frac{\eta}{2}\|\boldsymbol{h}(\boldsymbol{x})\|^2$$

for a positive value of the parameter η. This change gives a new Lagrangian function, the so-called *augmented Lagrangian*

$$\mathcal{L}_A(\boldsymbol{x}, \boldsymbol{u}) = \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) + \frac{\eta}{2}\|\boldsymbol{h}(\boldsymbol{x})\|^2$$

for which $(\boldsymbol{x}^*,\boldsymbol{u}^*)$ is a critical point. The Hessian of the the augmented Lagrangian at $(\boldsymbol{x}^*,\boldsymbol{u}^*)$ has the form

$$H\mathcal{L}_A(\boldsymbol{x}^*, \boldsymbol{u}^*) = H\mathcal{L}^* + \eta\nabla\boldsymbol{h}(\boldsymbol{x}^*)\nabla\boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}}.$$

It follows from **A4** that there exists a positive value η^* such that for $\eta \geq \eta^*$, $H\mathcal{L}_A(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is positive definite. If the value of η^* is known then the BFGS-SQP algorithm can be applied directly to the transformed problem with $\eta \geq \eta^*$ and, by Theorem 3.7, a superlinearly convergent algorithm results. This version of the SQP algorithm has been extensively studied and implemented. Although adequate for local convergence theory, this augmented-BFGS method has major drawbacks that are related to the fact that it is difficult to choose an appropriate value of η. To apply Theorem 3.7, the value of η must be large enough to ensure that $H\mathcal{L}_A(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is positive definite, a condition that requires *a priori* knowledge of $\boldsymbol{x}^*$. If unnecessarily large values of η are used without care numerical instabilities can result. This problem is exacerbated by the fact that if the iterates are not close to the solution appropriate values of η may not exist.

Quite recently, an intriguing adaptation of the BFGS-SQP algorithm has been suggested that shows promise of leading to a resolution of the above difficulties. This method, called the SALSA-SQP method is related to the augmented Lagrangian SQP method but differs (locally) in the sense that a precise estimate of η can be chosen independent of $\boldsymbol{x}^*$. If

$$\mathbf{y}_A = \nabla\mathcal{L}_A(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}) - \nabla\mathcal{L}_A(\boldsymbol{x}^k, \boldsymbol{u}^{k+1})$$

then

$$\mathbf{y}_A = \mathbf{y} + \eta\nabla\boldsymbol{h}(\boldsymbol{x}^{k+1})\boldsymbol{h}(\boldsymbol{x}^{k+1})$$

where $\mathbf{y}$ is the vector generated by the ordinary Lagrangian in (3.22). If

$$\mathbf{y}_A^{\mathrm{t}}\mathbf{s} > 0, \tag{3.26}$$

then the BFGS updates for the augmented Lagrangian have the hereditary positive definiteness property. A minimum value of η_k, not directly dependent on $\boldsymbol{x}^*$, can be given that guarantees that $\mathbf{y}_A^{\mathsf{t}}\mathbf{s}$ is "sufficiently" positive for given values of $\boldsymbol{x}^k$ and $\boldsymbol{u}^{k+1}$ in a neighborhood of $(\boldsymbol{x}^*,\boldsymbol{u}^*)$. The (local) version of this algorithm proceeds by using the augmented Lagrangian at each iteration with the required value of η_k (which may be zero) to force the satisfaction of this condition. The B_k generated by this procedure are positive definite while $H\mathcal{L}^*$ may not be; hence, the standard bounded deterioration property is not applicable. As a result a local convergence theorem is not yet available. As in the Powell-SQP algorithm, an R-superlinear rate of convergence of the $\boldsymbol{x}^k$ to $\boldsymbol{x}^*$ has been shown, provided that the sequence is known to converge.

3.4. Reduced Hessian SQP Methods

An entirely different approach to approximating the Hessian is based on the fact that assumption **A4** requires $H\mathcal{L}^*$ to be positive definite only on a particular subspace. The *reduced Hessian* methods approximate only the portion of the Hessian matrix relevant to this subspace. The advantages of these methods are that the standard positive definite updates can be used and that the dimension of the problem is reduced to $n - m$ (possibly a significant reduction). In this section we discuss the local convergence properties of such an approach. Several versions of a reduced Hessian type of algorithm have been proposed; they differ in the ways the multiplier vectors are chosen and the way the reduced Hessian approximation is updated; in particular, in the form of $\boldsymbol{y}$ that is used (see (3.31)). A general outline of the essential features of the method is given below.

Let $\boldsymbol{x}^k$ be an iterate for which $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$ has full rank and let Z_k and Y_k be matrices whose columns form bases for the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathsf{t}}$ and range space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$ respectively. Also assume that the columns of Z_k are orthogonal. Z_k and Y_k could be obtained, for example, by a QR factorization of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$.

Definition 3.2 Let $(\boldsymbol{x}^k,\boldsymbol{u}^k)$ be a given solution-multiplier pair and assume that $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$ has full rank. The matrix

$$Z_k{}^{\mathsf{t}} H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k) Z_k$$

is called a **reduced Hessian** for the Lagrangian function at $(\boldsymbol{x}^k,\boldsymbol{u}^k)$.

The reduced Hessian is not unique; its form depends upon the choice of basis for the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathsf{t}}$. Since by **A4** a reduced Hessian at the solution is positive definite, it follows that if $(\boldsymbol{x}^k,\boldsymbol{u}^k)$ is close enough to $(\boldsymbol{x}^*,\boldsymbol{u}^*)$ then the reduced Hessian at $(\boldsymbol{x}^k, \boldsymbol{u}^k)$ is positive definite. Decomposing the vector

$\boldsymbol{d}_x$ as

$$\boldsymbol{d}_x = Z_k\,\mathbf{p}_Z + Y_k\,\mathbf{p}_Y \tag{3.27}$$

it can be seen that the constraint equation of (ECQP) becomes

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathsf{t}} Y_k\,\mathbf{p}_Y = -\boldsymbol{h}(\boldsymbol{x}^k)$$

which by virtue of **A2** can be solved to obtain

$$\mathbf{p}_Y = -[\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathsf{t}} Y_k]^{-1} \boldsymbol{h}(\boldsymbol{x}^k). \tag{3.28}$$

The minimization problem (ECQP) is now an unconstrained problem in $n - m$ variables given by

$$\underset{\mathbf{p}_Z}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{p}_Z{}^{\mathsf{t}} Z_k{}^{\mathsf{t}} B_k Z_k\,\mathbf{p}_Z + (\nabla f(\boldsymbol{x}^k)^{\mathsf{t}} + \mathbf{p}_Y{}^{\mathsf{t}} B_k) Z_k\,\mathbf{p}_Z.$$

The matrix in this unconstrained problem is an approximation to the reduced Hessian of the Lagrangian at $\boldsymbol{x}^k$. Rather than update B_k and then compute $Z_k{}^{\mathsf{t}} B_k Z_k$, the reduced matrix itself is updated. Thus at a particular iteration, given a positive definite $(n - m) \times (n - m)$ matrix R_k, an iterate $\boldsymbol{x}^k$ and a multiplier estimate $\boldsymbol{u}^k$, the new iterate can be found by first computing $\mathbf{p}_Y$ from (3.28), setting

$$\mathbf{p}_Z = -R_k^{-1} Z_k{}^{\mathsf{t}} (\nabla f(\boldsymbol{x}^k) + B_k\,\mathbf{p}_Y), \tag{3.29}$$

and setting $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{d}_x$, where $\boldsymbol{d}_x$ is given by (3.27). A new multiplier, $\boldsymbol{u}^{k+1}$, is generated and the reduced Hessian approximation is updated using the BFGS formula (3.23) with the R_k replacing B_k,

$$\boldsymbol{s} = Z_k{}^{\mathsf{t}} (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = Z_k{}^{\mathsf{t}} Z_k\,\mathbf{p}_Z \tag{3.30}$$

and

$$\boldsymbol{y} = Z_k{}^{\mathsf{t}} [\nabla \mathcal{L}(\boldsymbol{x}^k + Z_k\,\mathbf{p}_Z, \boldsymbol{u}^k) - \nabla \mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k)]. \tag{3.31}$$

The choices of $\boldsymbol{s}$ and $\boldsymbol{y}$ are motivated by the fact that only the reduced Hessian is being approximated.

This method does not stipulate a new multiplier iterate directly since the problem being solved at each step is unconstrained. However, the least squares solution for the first order conditions (cf. (3.2)) can be used. Generally, all that is needed is that the multipliers satisfy

$$\left\| \boldsymbol{u}^k - \boldsymbol{u}^* \right\| = \mathcal{O}\left(\left\| \boldsymbol{x}^k - \boldsymbol{x}^* \right\| \right). \tag{3.32}$$

Since $\mathcal{P}^k = Z_k Z_k{}^{\mathsf{t}}$ the approximation

$$Z_k R_k Z_k{}^{\mathsf{t}}$$

can be thought of as an approximation of

$$\mathcal{P}^k H\mathcal{L}(\boldsymbol{x}^k, \boldsymbol{u}^k) \mathcal{P}^k.$$

Thus since this method does not approximate

$$\mathcal{P}^k H\mathcal{L}^*$$

neither the local convergence theorem nor the superlinear rate of convergence theorem, Theorems 3.2 and 3.4, follow as for full Hessian approximations. Nevertheless, the two-sided approximation of the Hessian matrix suggests that the conditions of Theorem 3.5 may hold. In fact, if it is assumed that the matrices Z_k are chosen in a smooth way, i.e., so that

$$\|Z_k - Z(x^*)\| = \mathcal{O}\left(\left\|x^k - x^*\right\|\right), \tag{3.33}$$

the assumption of local convergence leads to two-step superlinear convergence.

Theorem 3.8 *Assume that the reduced Hessian algorithm is applied with u^k and Z_k chosen so that (3.32) and (3.33) are satisfied. If the sequence $\{x^k\}$ converges to x^* R-linearly then $\{R_k\}$ and $\{R_k^{-1}\}$ are uniformly bounded and $\{x^k\}$ converges two-step superlinearly.*

While the condition on the multiplier iterates are easy to satisfy, some care is required to ensure that (3.33) is satisfied as it has been observed that arbitrary choices of Z_k may be be discontinuous and hence invalidate the theorem.

3.5. Notes and References

The study of Newton's method applied to the first order necessary conditions for constrained optimization can be found in (Tapia 1974) and (Goodman 1985). The equivalence of Newton's method applied to the system of first order equations and feasibility conditions and the SQP step with the Lagrangian Hessian was first mentioned by Tapia (1978).

The expository paper, (Dennis and Moré 1977), is a good source for an introductory discussion of bounded deterioration and of updating methods satisfying the secant condition in the setting of unconstrained optimization. The paper by Broyden, Dennis and Moré (1973) provides the basic results on the convergence properties for quasi-Newton methods.

Theorems (3.2) and (3.4) were first proven by Boggs, Tolle and Wang (1982). The latter theorem was proven under the assumption that the iterates converge linearly, generalizing the result for unconstrained optimization given in Dennis and Moré (1977). The assumption was substantially weakened by several authors; see, for example, (Fontecilla, Steihaug and Tapia 1987). Theorem (3.5) is due to Powell (1978*b*). The paper by Coleman (1990) contains a good general overview of superlinear convergence theorems for SQP methods.

The (local) superlinear convergence of the PSB-SQP method was proven by Han (1976) in his first paper on SQP methods. This paper also included the proof of superlinear covergence for the positive definite updates when the Hessian of the Lagrangian is positive definite (Theorem (3.7)) and suggested the use of the augmenting term in the general case. The augmented method was also investigated in (Schittkowski 1981) and (Tapia 1977). A description of the Powell-SQP method appears in (Powell 1978*b*) while the Salsa-SQP method is introduced in (Tapia 1988) and developed further in (Byrd, Tapia and Zhang 1992).

The reduced Hessian SQP method has been the subject of research for a number of authors. The presentation here follows that of Byrd and Nocedal (1991). Other important articles include (Coleman and Conn 1984), (Nocedal and Overton 1985), (Gabay 1982), (Yuan 1985), and (Gilbert 1993). The problems involved in satisfying (3.33) have been addressed in (Byrd and Schnabel 1986) , (Coleman and Sorensen 1984), and (Gill, Murray, Saunders, Stewart and Wright 1985).

Other papers of interest on local convergence for SQP methods include (Schittkowski 1983), (Panier and Tits 1993), (Bertsekas 1980), (Coleman and Feynes 1992), (Glad 1979), (Fukushima 1986), and (Fontecilla 1988).

4. Merit Functions and Global Convergence

In the previous section we demonstrated the existence of variants of the SQP method that are rapidly locally convergent. Here we show how the merit function assures that the iterates eventually get close to a critical point; in the next section we point out some gaps in the theory that prevent a complete analysis.

A *merit function* ϕ is incorporated into an SQP algorithm for the purpose of achieving global convergence. As described in Section 2.3, a line search procedure is used to modify the length of the step $\boldsymbol{d}_x$ so that the step from $\boldsymbol{x}^k$ to $\boldsymbol{x}^{k+1}$ reduces the value of ϕ. This reduction is taken to imply that acceptable progress towards the solution is being made.

The standard way to ensure that a reduction in ϕ indicates progress is to construct ϕ so that the solutions of (NLP) are the unconstrained minimizers of ϕ. Then it must be possible to decrease ϕ by taking a step in the direction $\boldsymbol{d}_x$ generated by solving the quadratic subproblem, i.e., $\boldsymbol{d}_x$ must be a descent direction for ϕ. If this is the case, then for a sufficiently small α, $\phi(\boldsymbol{x}^k+\alpha\boldsymbol{d}_x)$ will be less than $\phi(\boldsymbol{x}^k)$. An appropriate steplength that decreases ϕ can then be computed; for example, by a "backtracking" procedure of trying successively smaller values of α until a suitable one is obtained.

Given that a merit function is found that has these properties and that a procedure is used to take a step that decreases the function, global convergence proofs for the resulting SQP algorithm are generally similar to those found in unconstrained optimization: they depend on proving that the limit points of the $\boldsymbol{x}$-iterates are critical points of ϕ. These proofs rely heavily on being able to guarantee that a "sufficient" decrease in ϕ can be achieved at each iteration..

In unconstrained minimization there is a natural merit function; namely, the objective function itself. In the constrained setting, unless the iterates are always feasible, a merit function has to balance the drive to decrease f with the need to satisfy the constraints. This balance is often controlled by a parameter in ϕ that weights a measure of the infeasibility against the value of either the objective function or the Lagrangian function. In this section, we illustrate these ideas by describing two of the most popular merit functions: a differentiable augmented Lagrangian function and the simpler, but nondifferentiable, ℓ_1 penalty function. We derive some of the important properties of these functions and discuss some of the advantages and disadvantages of each. In addition, we provide appropriate rules for choosing the steplength parameter so as to obtain the basic convergence results.

To simplify the presentation we restrict our attention at the beginning to the problem with only equality constraints. Extensions that allow inequality constraints are discussed at the ends of the sections.

As might be expected, some additional assumptions are needed if global convergence is to be achieved. In fact, the need for these assumptions raises the issue as to what exactly is meant by "global" convergence. It is impossible to conceive of an algorithm that would converge from any starting point for every (NLP). The very nature of nonlinearity allows the possibility, for example, of perfectly satisfactory iterations following a steadily decreasing function towards achieving feasibility and a minimum at some infinite point. In order to focus attention on the methods and not the problem structure we make the following assumptions.

C1: The starting point and all succeeding iterates lie in some compact set $\mathcal{C}$.

C2: The columns of $\nabla \boldsymbol{h}(\boldsymbol{x})$ are linearly independent for all $\boldsymbol{x} \in \mathcal{C}$.

The first assumption is made in some guise in almost all global convergence proofs, often by making specific assumptions about the functions. The second assumption assures that the systems of linearized constraints are consistent. An additional assumption will be made about the matrix approximations B_k, depending on the particular merit function.

4.1. Augmented Lagrangian merit functions

Our first example of a merit function is the augmented Lagrangian function. There are several versions of this function; we use the following version to illustrate the class:

$$\phi_F(\boldsymbol{x};\eta) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\overline{\boldsymbol{u}}(\boldsymbol{x}) + \frac{\eta}{2}\|\boldsymbol{h}(\boldsymbol{x})\|_2^2 \tag{4.1}$$

where η is a constant to be determined and

$$\overline{\boldsymbol{u}}(\boldsymbol{x}) = -\left[\nabla \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\nabla \boldsymbol{h}(\boldsymbol{x})\right]^{-1}\nabla \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\nabla f(\boldsymbol{x}). \tag{4.2}$$

The multipliers $\overline{\boldsymbol{u}}(\boldsymbol{x})$ defined by (4.2) are the least squares estimates of the optimal multipliers based on the first order necessary conditions and hence $\overline{\boldsymbol{u}}(\boldsymbol{x}^*) = \boldsymbol{u}^*$ (see (3.1) and (3.2)). Under the assumptions, ϕ_F and $\overline{\boldsymbol{u}}$ are differentiable and ϕ_F is bounded below on $\mathcal{C}$ for η sufficiently large. The following formulas will be useful in the discussion:

$$\begin{aligned}
\nabla\overline{\boldsymbol{u}}(\boldsymbol{x}) &= -H\mathcal{L}(\boldsymbol{x},\overline{\boldsymbol{u}}(\boldsymbol{x}))\nabla\boldsymbol{h}(\boldsymbol{x})\left[\nabla \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\nabla \boldsymbol{h}(\boldsymbol{x})\right]^{-1} \\
&\quad + R_1(\boldsymbol{x}) && (4.3)\\
\nabla\phi_F(\boldsymbol{x};\eta) &= \nabla f(\boldsymbol{x}) + \nabla\boldsymbol{h}(\boldsymbol{x})\overline{\boldsymbol{u}}(\boldsymbol{x}) \\
&\quad + \nabla\overline{\boldsymbol{u}}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}) + \eta\nabla\boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}). && (4.4)
\end{aligned}$$

The function $R_1(\boldsymbol{x})$ in (4.3) is bounded and satisfies $R_1(\boldsymbol{x}^*) = \mathbf{0}$ if $\boldsymbol{x}^*$ satisfies the first order necessary conditions.

The following theorem establishes that the augmented Lagrangian merit function has the essential properties of a merit function for the equality-constrained nonlinear program.

Theorem 4.1 *Assume* **B1**, **B2**, **C1**, *and* **C2** *are satisfied. Then for η sufficiently large the following hold:*

(i) $\boldsymbol{x}^* \in \mathcal{C}$ is a strong local minimum of ϕ_F if and only if $\boldsymbol{x}^*$ is a strong local minimum of (NLP) and

(ii) if $\boldsymbol{x}$ is not a critical point of (NLP) then $\boldsymbol{d}_x$ is a descent direction for ϕ_F.

The proof is rather technical, but it does illustrate the techniques that are often used in such arguments and it provides a useful intermediate result, namely (4.10). A key idea is to decompose certain vectors according to the range and null space projections $\mathcal{P}$ and $\mathcal{Q}$.

Proof. If $\boldsymbol{x}^* \in \mathcal{C}$ is feasible, and satisfies the first order necessary conditions, then it follows from (4.4) that $\nabla\phi_F(\boldsymbol{x}^*,\eta) = 0$. Conversely, if $\boldsymbol{x}^* \in \mathcal{C}$, $\nabla\phi_F(\boldsymbol{x}^*;\eta) = 0$, and η is sufficiently large then it follows from **C1** and **C2** that $\boldsymbol{h}(\boldsymbol{x}^*) = 0$ and $\boldsymbol{x}^*$ satisfies **A1** for (NLP). To establish (i), the relation between $H\phi_F(\boldsymbol{x}^*;\eta)$ and $H\mathcal{L}^*$ must be explored for such points $\boldsymbol{x}^*$. It follows from (4.4) that

$$H\phi_F(\boldsymbol{x}^*;\eta) = H\mathcal{L}^* - \mathcal{Q}^*H\mathcal{L}^* - H\mathcal{L}^*\mathcal{Q}^* + \eta\nabla\boldsymbol{h}(\boldsymbol{x}^*)\nabla\boldsymbol{h}(\boldsymbol{x}^*)^{\mathrm{t}} \tag{4.5}$$

where $\mathcal{Q}^*$ is defined by (2.3). Now let $y \in \mathcal{R}^n$, $y \neq 0$, be arbitrary. Then setting $y = \mathcal{Q}^* y + \mathcal{P}^* y$, (4.5) yields

$$\begin{array}{rcl} y^{\mathrm{t}} H\phi_F(x^*;\eta) y & = & (\mathcal{P}^* y)^{\mathrm{t}} H\mathcal{L}^*(\mathcal{P}^* y) - (\mathcal{Q}^* y)^{\mathrm{t}} H\mathcal{L}^*(\mathcal{Q}^* y) \\ & & +\eta(\mathcal{Q}^* y)^{\mathrm{t}} \left[\nabla h(x^*)\nabla h(x^*)^{\mathrm{t}}\right](\mathcal{Q}^* y). \end{array} \tag{4.6}$$

Assume that x^* is a strong local minimum of (NLP). Since $\mathcal{Q}^* y$ is in the range space of $\nabla h(x^*)$, it follows from **A2** that there exists a constant μ such that

$$(\mathcal{Q}^* y)^{\mathrm{t}} \left[\nabla h(x^*)\nabla h(x^*)^{\mathrm{t}}\right](\mathcal{Q}^* y) \geq \mu \left\|\mathcal{Q}^* y\right\|^2.$$

Let the maximum eigenvalue of $H\mathcal{L}^*$ in absolute value be σ_{max} and let σ_{min} be the minimum eigenvalue of $\mathcal{P}^* H\mathcal{L}^*\mathcal{P}^*$, which by **A4** is positive. Defining

$$\epsilon = \frac{\|\mathcal{Q}^* y\|}{\|y\|},$$

which implies

$$\frac{\|\mathcal{P}^* y\|^2}{\|y\|^2} = 1 - \epsilon^2,$$

and dividing both sides of (4.6) by $\|y\|^2$ gives

$$\frac{y^{\mathrm{t}} H\phi_F(x^*;\eta) y}{\|y\|^2} \geq \sigma_{min} + (\eta\mu - \sigma_{max} - \sigma_{min})\epsilon^2.$$

This last expression will be positive for η sufficiently large, which implies that x^* is a strong local minimum of ϕ_F. Conversely, if x^* is a strong local minimum of ϕ_F then (4.6) must be positive for all y. This implies that $H\mathcal{L}^*$ must be positive definite on the null space of $\nabla h(x^*)^{\mathrm{t}}$ and hence that x^* is a strong local minimum of (NLP). This establishes (i).

To show that the direction d_x is always a descent direction for ϕ_F, the inner product of both sides of (4.4) is taken with d_x to yield

$$\begin{array}{rcl} {d_x}^{\mathrm{t}}\nabla\phi_F(x^k;\eta) & = & {d_x}^{\mathrm{t}}\nabla f(x^k) + {d_x}^{\mathrm{t}}\nabla h(x^k)\overline{u}(x^k) \\ & & +{d_x}^{\mathrm{t}}\nabla\overline{u}(x^k)h(x^k) + \eta {d_x}^{\mathrm{t}}\nabla h(x^k)h(x^k). \end{array} \tag{4.7}$$

Using (4.2) it follows that

$$\begin{array}{rcl} {d_x}^{\mathrm{t}}\nabla h(x^k)\overline{u}(x^k) & = & -{d_x}^{\mathrm{t}}\nabla h(x^k)\left[\nabla h(x)^{\mathrm{t}}\nabla h(x)\right]^{-1}\nabla h(x)^{\mathrm{t}}\nabla f(x) \\ & = & -{d_x}^{\mathrm{t}}\mathcal{Q}^k\nabla f(x^k). \end{array} \tag{4.8}$$

Writing $d_x = \mathcal{Q}^k d_x + \mathcal{P}^k d_x$ and noting that $h(x^k) = -\nabla h(x^k)^{\mathrm{t}}\mathcal{Q}^k d_x$ results in

$$\begin{array}{rcl} {d_x}^{\mathrm{t}}\nabla\phi_F(x^k;\eta) & = & \mathcal{P}^k {d_x}^{\mathrm{t}}\nabla f(x^k) - {d_x}^{\mathrm{t}}\nabla\overline{u}(x^k)\nabla h(x^k)^{\mathrm{t}}\mathcal{Q}^k d_x \\ & & -\eta(\mathcal{Q}^k d_x)^{\mathrm{t}}\left[\nabla h(x^k)\nabla h(x^k)^{\mathrm{t}}\right](\mathcal{Q}^k d_x). \end{array}$$

Finally, using the first order necessary conditions for (ECQP) and the fact that $\mathcal{P}^k \nabla h(x^k)^{\mathrm{t}} = 0$, the following is obtained:

$$\begin{aligned} {d_x}^{\mathrm{t}} \nabla \phi_F(x^k;\eta) &= -(\mathcal{P}^k d_x)^{\mathrm{t}} B_k (\mathcal{P}^k d_x) - (\mathcal{Q}^k d_x)^{\mathrm{t}} B_k d_x \\ &\quad - {d_x}^{\mathrm{t}} \nabla \overline{u}(x^k) \nabla h(x^k)^{\mathrm{t}} \mathcal{Q}^k d_x \\ &\quad -\eta (\mathcal{Q}^k d_x)^{\mathrm{t}} \left[\nabla h(x^k) \nabla h(x^k)^{\mathrm{t}} \right] (\mathcal{Q}^k d_x). \end{aligned} \tag{4.9}$$

Now, from (4.3) and assumptions **C1** and **C2** it follows that

$${d_x}^{\mathrm{t}} \nabla \overline{u}(x^k) \nabla h(x^k)^{\mathrm{t}} \mathcal{Q}^k d_x \le \gamma_1 \left\| \mathcal{Q}^k d_x \right\| \|d_x\|$$

for some constant γ_1. Dividing both sides of (4.9) by $\|d_x\|^2$, letting

$$\epsilon = \frac{\left\| \mathcal{Q}^k d_x \right\|}{\|d_x\|},$$

and using **B1** yields

$$\frac{\nabla \phi_F(x^k;\eta)^{\mathrm{t}} d_x}{\|d_x\|^2} \le -\beta_1 + \gamma_2 \epsilon - \eta \gamma_3 \epsilon^2 \tag{4.10}$$

for constants γ_2 and γ_3. The quantity on the left of (4.10) is then negative and can be uniformly bounded away from zero provided η is sufficiently large, thus proving d_x is a descent direction for ϕ. □

It is interesting to observe that the value of η necessary to obtain descent of d_x depends on the eigenvalue bounds on $\{B_k\}$, whereas the value of η necessary to ensure that x^* is a strong local minimizer of ϕ_F depends on the eigenvalues of $H\mathcal{L}^*$. Thus a strategy to adjust η to achieve a good descent direction may not be sufficient to prove that $H\phi_F(x^*)$ is positive definite. We comment further on this below.

This line of reasoning can be continued to obtain a more useful form of the descent result. From (4.4) and arguments similar to those above,

$$\left\| \nabla \phi_F(x^k;\eta) \right\| \le \gamma_4 \|d_x\|$$

for some constant γ_4. Thus from (4.10) it follows that there exists a constant γ_5 such that

$$\frac{\nabla \phi_F(x^k;\eta)^{\mathrm{t}} d_x}{\left\| \nabla \phi_F(x^k;\eta) \right\| \|d_x\|} \le -\gamma_5 < 0 \tag{4.11}$$

for all k. This inequality ensures that the cosine of the angle θ_k between the direction d_x and the negative gradient of ϕ_F is bounded away from zero, i.e., for all k

$$\cos(\theta_k) = -\frac{\nabla \phi_F(x^k;\eta)^{\mathrm{t}} d_x}{\left\| \nabla \phi_F(x^k;\eta) \right\| \|d_x\|} \ge \gamma_5 > 0. \tag{4.12}$$

This condition is sufficient to guarantee convergence of the sequence if it is coupled with suitable line search criteria that impose restrictions on the steplength α. For example, the Wolfe conditions require a steplength α to satisfy

$$\begin{aligned} \phi_F(\boldsymbol{x}^k + \alpha \boldsymbol{d}_x; \eta) &\leq \phi_F(\boldsymbol{x}^k; \eta) + \sigma_1 \alpha \nabla \phi_F(\boldsymbol{x}^k; \eta)^{\mathrm{t}} \boldsymbol{d}_x \quad (4.13) \\ \nabla \phi_F(\boldsymbol{x}^k + \alpha \boldsymbol{d}_x; \eta)^{\mathrm{t}} \boldsymbol{d}_x &\geq \sigma_2 \nabla \phi_F(\boldsymbol{x}^k; \eta)^{\mathrm{t}} \boldsymbol{d}_x \quad (4.14) \end{aligned}$$

where $0 < \sigma_1 < \sigma_2 < 1$. Inequality (4.13) ensures that there will be a sufficient reduction in ϕ_F while (4.14) guarantees that the steplength α will not be too small. An important property of these conditions is that if $\boldsymbol{d}_x$ is a descent direction for ϕ_F, then a steplength α satisfying (4.13)–(4.14) can always be found. Furthermore, the reduction in ϕ_F for such a step satisfies

$$\phi_F(\boldsymbol{x}^{k+1}; \eta) \leq \phi_F(\boldsymbol{x}^k; \eta) - \gamma_6 \cos^2(\theta_k) \left\| \nabla \phi_F(\boldsymbol{x}^k; \eta) \right\|^2,$$

for some positive constant γ_6. Therefore

$$\sum_{k=1}^{\infty} \cos^2(\theta_k) \left\| \nabla \phi_F(\boldsymbol{x}^k; \eta) \right\|^2 < \infty \quad (4.15)$$

and, since $\cos(\theta_k)$ is uniformly bounded away from 0, it follows that

$$\lim_{k \to \infty} \nabla \phi_F(\boldsymbol{x}^k; \eta) = 0.$$

This implies that $\{\boldsymbol{x}^k\}$ converges to a critical point of ϕ_F. The following theorem results:

Theorem 4.2 *Assume that η is chosen such that* (4.12) *holds. Then the SQP algorithm with steplength α chosen to satisfy the Wolfe conditions* (4.13)–(4.14) *is globally convergent to a critical point of ϕ_F.*

If the critical points of ϕ_F all correspond to local minima of (NLP) then the algorithm will converge to a local solution. This, however, can rarely be guaranteed. Only convergence to a critical point, and not to a local minimum of ϕ_F, is guaranteed. It is easy to see from the following example that convergence can occur to a local maximum of (NLP). Consider

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \boldsymbol{x}_1 \\ &\text{subject to: } \boldsymbol{x}_1^2 + \boldsymbol{x}_2^2 - 1 = 0 \end{aligned} \quad (4.16)$$

with a starting approximation of $(2, 0)$ and $B_k = I$ for all k. The iterates will converge from the right to the point $(1, 0)$, which is a local maximum of the problem, but ϕ_F will decrease appropriately at each iteration. The point $(1, 0)$, however, is a saddle point of ϕ_F. In practice convergence to a maximizer of (NLP) rarely occurs.

Theorems 4.2 and 4.1 require that η be sufficiently large. Since it is not known in advance how large η needs to be, it is necessary to employ an adaptive strategy for adjusting η when designing a practical code. Such adjustment strategies can have a dramatic effect on the performance of the implementation, as is discussed in Section 7.

Augmented Lagrangian merit functions have been extended to handle inequality constrained problems in several ways. Two successful approaches are described below.

In the first an *active set* strategy is employed, i.e., at each iteration a set of the inequality constraints is selected and treated as if they were equality constraints; the remaining inequalities are handled differently. The active set is selected by using the multipliers from (QP). In particular,

$$\mathcal{I}_k = \left\{ i \ : \ g_i(\boldsymbol{x}^k) \geq -\tfrac{(\boldsymbol{v}_{qp})_i}{\eta} \right\}.$$

With this choice, $\mathcal{I}_k$ will contain all unsatisfied constraints and no "safely satisfied" ones. The merit function at $\boldsymbol{x}^k$ is defined by

$$\begin{aligned}\phi_{FI}(\boldsymbol{x}, \boldsymbol{u}_{qp}, \boldsymbol{v}_{qp}; \eta) \quad = \quad & f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}} \boldsymbol{u}_{qp}(\boldsymbol{x}) + \tfrac{1}{2}\eta \, \|\boldsymbol{h}(\boldsymbol{x})\|^2 \\ & + \sum_{i \in \mathcal{I}_k} \left(\boldsymbol{g}_i(\boldsymbol{x})(\boldsymbol{v}_{qp}(\boldsymbol{x}))_i + \tfrac{1}{2}\eta \boldsymbol{g}_i(\boldsymbol{x})^2 \right) \\ & + \frac{1}{2\eta} \sum_{i \notin \mathcal{I}_k} (\boldsymbol{v}_{qp}(\boldsymbol{x}))_i^2 .\end{aligned}$$

ϕ_{FI} is still differentiable and, as a result of **A3**, the correct active set is eventually identified in a neighborhood of the solution. In this formulation, the multipliers are the multipliers from (QP), not the least squares multipliers. Therefore ϕ_{FI} is a function of both $\boldsymbol{x}$ and these multipliers and, consequently, the analysis for ϕ_{FI} is somewhat more complicated.

A second approach uses the idea of squared slack variables. One can consider the problem

$$\begin{array}{lrcl} \underset{\boldsymbol{x}, \boldsymbol{t}}{\text{minimize}} & f(\boldsymbol{x}) & & \\ \text{subject to:} & \boldsymbol{h}(\boldsymbol{x}) & = & 0 \\ & \boldsymbol{g}_i(\boldsymbol{x}) + (\boldsymbol{t}_i)^2 & = & 0 \\ & i = 1, \ldots, p & & \end{array} \tag{4.17}$$

where $\boldsymbol{t}$ is the vector of slack variables. This problem is equivalent to (NLP) in the sense that both have a strong local solution at $\boldsymbol{x}^*$ where, in (4.17), $(\boldsymbol{t}_i^*)^2 = -\boldsymbol{g}_i(\boldsymbol{x}^*)$. By writing out the function ϕ_F for the problem (4.17), it is observed that the merit function only involves the squares of $\boldsymbol{t}_i$. Thus, by letting $\boldsymbol{z}_i = \boldsymbol{t}_i^2$ and setting

$$\overline{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{z}) = \begin{pmatrix} \boldsymbol{h}(\boldsymbol{x}) \\ \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{z} \end{pmatrix},$$

we can construct the merit function

$$\phi_{FZ}(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x}) + \overline{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{z})^{\mathrm{t}}\overline{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{z}) + \frac{\eta}{2}\left\|\overline{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{z})\right\|^2.$$

Here $\overline{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{z})$ is the least squares estimate for all of the multipliers. Note that (4.17) is not used to create the quadratic subproblem, but rather (QP) is solved at each iteration to obtain the step $\boldsymbol{d}_x$. The slack variables can then be updated at each iteration in a manner guaranteed to maintain the nonnegativity of $\boldsymbol{z}$. For example, a step

$$\boldsymbol{d}_z = -(\nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{g}(\boldsymbol{x}^k) + \boldsymbol{z}^k)$$

can be calculated. Then the constraints of (QP) imply that $\boldsymbol{z}^{k+1} = \boldsymbol{z}^k + \alpha \boldsymbol{d}_z \geq 0$ if $\boldsymbol{z}^k \geq 0$ and $\alpha \in (0, 1]$.

4.2. The ℓ_1 Merit Function

One of the first merit functions to be introduced was the ℓ_1 exact penalty function that, in the equality constrained case, is given by

$$\phi_1(\boldsymbol{x};\rho) = f(\boldsymbol{x}) + \rho\left\|\boldsymbol{h}(\boldsymbol{x})\right\|_1 \tag{4.18}$$

where ρ is a positive constant to be chosen. The properties of this function *vis-a-vis* the equality-constrained optimization problem have been well-documented in the literature. For our purposes it is sufficient to note that ϕ_1, like the augmented Lagrangian of the previous section, is an "exact" penalty function; that is, there exists a positive ρ^* such that for all $\rho \geq \rho^*$, an unconstrained minimum of ϕ_1 corresponds to a solution of (NLP). Note that ϕ_1 is not differentiable on the feasible set. If the penalty term were squared to achieve differentiability then the "exact" property would be lost; minimizers of ϕ_1 would only converge to solutions of (NLP) as $\rho \to \infty$.

Although ϕ_1 is not differentiable, it does have a directional derivative along $\boldsymbol{d}_x$. It can be shown that this directional derivative, denoted by the operator D, is given by

$$D(\phi_1(\boldsymbol{x}^k;\rho);\boldsymbol{d}_x) = \nabla f(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x - \rho\left\|h(\boldsymbol{x}^k)\right\|_1. \tag{4.19}$$

Substituting the first order necessary conditions for (ECQP) into (4.19) yields

$$D(\phi_1(\boldsymbol{x}^k;\rho);\boldsymbol{d}_x) = -\boldsymbol{d}_x{}^{\mathrm{t}}B_k\boldsymbol{d}_x - \boldsymbol{d}_x{}^{\mathrm{t}}\nabla \boldsymbol{h}(\boldsymbol{x}^k)\boldsymbol{u}_{qp} - \rho\left\|\boldsymbol{h}(\boldsymbol{x}^k)\right\|_1.$$

It follows from the linearized constraints of (ECQP) that

$$\boldsymbol{d}_x{}^{\mathrm{t}}\nabla \boldsymbol{h}(\boldsymbol{x}^k)\boldsymbol{u}_{qp} = -\boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{u}_{qp}$$

and, since

$$\boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{u}_{qp} \leq \left\|\boldsymbol{u}_{qp}\right\|_\infty \left\|\boldsymbol{h}(\boldsymbol{x}^k\right\|_1, \tag{4.20}$$

the inequality

$$D(\phi(\boldsymbol{x}^k;\rho);\boldsymbol{d}_x) \leq -\boldsymbol{d}_x{}^{\mathrm{t}} B_k \boldsymbol{d}_x - (\rho - \|\boldsymbol{u}_{qp}\|_\infty)\left\|\boldsymbol{h}(\boldsymbol{x}^k)\right\|_1 \tag{4.21}$$

is obtained. In order to have $\boldsymbol{d}_x$ be a descent direction for ϕ_1 and to obtain a convergence theorem it is sufficient to assume the uniform positive definiteness of $\{B_k\}$:

B4: For all $\mathbf{d} \in \mathcal{R}^n$ there are positive constants β_1 and $\beta_2 > 0$ such that

$$\beta_1 \|\mathbf{d}\|^2 \leq \mathbf{d}^{\mathrm{t}} B_k \mathbf{d} \leq \beta_2 \|\mathbf{d}\|^2$$

for all k.

The assumptions **B1** and **B2** that were made for the augmented Lagrangian are sufficient, but we make the stronger assumption to simplify the presentation. Using **B4**, the first term in (4.21) is always negative and thus $\boldsymbol{d}_x$ is a guaranteed descent direction for ϕ_1 if $\rho > \|\boldsymbol{u}_{qp}\|_\infty$.

Global convergence of an algorithm that uses ϕ_1 as a merit function can be demonstrated using arguments similar to those for Theorem 4.1. First, at each iteration, the parameter ρ is chosen by

$$\rho = \|\boldsymbol{u}_{qp}\|_\infty + \bar{\rho} \tag{4.22}$$

for some constant $\bar{\rho} > 0$. Next, the first Wolfe line search condition, (4.13), is replaced by

$$\phi_1(\boldsymbol{x}^k + \alpha \boldsymbol{d}_x) \leq \phi_1(\boldsymbol{x}^k) + \zeta\alpha D(\phi_1(\boldsymbol{x}^k;\rho);\boldsymbol{d}_x) \tag{4.23}$$

where $\zeta \in (0, \frac{1}{2})$. (The condition $\zeta < \frac{1}{2}$ is for technical reasons; in practice ζ is usually chosen to be much smaller.) Recall that the second of the Wolfe conditions, (4.14), is to prevent a steplength that is too small. The assumptions that have been made guarantee that the use of a backtracking line search procedure produces steplengths that are uniformly bounded away from zero for all iterations. Denoting this lower bound by $\bar{\alpha}$, it now follows from (4.21) and (4.23) that the reduction in ϕ_1 at each step satisfies

$$\phi_1(\boldsymbol{x}^k + \alpha \boldsymbol{d}_x) - \phi_1(\boldsymbol{x}^k) \leq -\zeta\bar{\alpha}\left[\mu_1 \|\boldsymbol{d}_x\|^2 + \bar{\rho}\left\|\boldsymbol{h}(\boldsymbol{x}^k)\right\|_1\right]. \tag{4.24}$$

Assumption **C1** and (4.24) imply that

$$\sum_{k=1}^{\infty}\left[\left\|\boldsymbol{d}_x(\boldsymbol{x}^k)\right\|^2 + \left\|\boldsymbol{h}(\boldsymbol{x}^k)\right\|_1\right] \leq \infty$$

and therefore $\left[\left\|\boldsymbol{d}_x(\boldsymbol{x}^k)\right\|^2 + \left\|\boldsymbol{h}(\boldsymbol{x}^k)\right\|_1\right] \to 0$ as $k \to \infty$. Since $\boldsymbol{d}_x = \mathbf{0}$ if and only if $\boldsymbol{x}^k$ is a feasible point satisfying **A1**, the following theorem results.

Theorem 4.3 *Assume that ρ is chosen such that* (4.22) *holds. Then the*

SQP algorithm started at any point $\boldsymbol{x}^0$ *with steplength* $\alpha \geq \bar{\alpha} > 0$ *chosen to satisfy* (4.23) *converges to a stationary point of* ϕ_1.

As in the case of ϕ_F convergence to a local minimum of (NLP) cannot be guaranteed. In fact, the same counterexample used in that case applies here.

An advantage of ϕ_1 over the augmented Lagrangian is that ϕ_1 is easily extended to handle inequalities. Since differentiability is not an issue in this case, the function ϕ_1 can simply be defined as

$$\phi_1(\boldsymbol{x}) = f(\boldsymbol{x}) + \rho \left[\|\boldsymbol{h}(\boldsymbol{x})\|_1 + \|\boldsymbol{g}^+(\boldsymbol{x})\|_1 \right]$$

where

$$\boldsymbol{g}^+(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \ \boldsymbol{g}_i(\boldsymbol{x}) \leq 0 \\ \boldsymbol{g}_i(\boldsymbol{x}) & \text{if } \ \boldsymbol{g}_i(\boldsymbol{x}) > 0 \end{cases}$$

All of the theoretical results continue to hold under this extension.

The merit function ϕ_1 has been popular because of its simplicity — it requires only the evaluation of f, $\boldsymbol{h}$, and $\boldsymbol{g}$ to check a prospective point. ϕ_F, on the other hand, is expensive to evaluate in that it requires the evaluation of f, $\boldsymbol{h}$, $\boldsymbol{g}$ and their gradients just to check a prospective point. In addition, in the large scale case, the evaluation of $\overline{\boldsymbol{u}}(\boldsymbol{x})$ involves nontrivial algebra. Implementations that use these merit functions partially circumvent this difficulty by using an approximation to ϕ_F, e.g., by approximating $\overline{\boldsymbol{u}}$ by a linear function or by keeping it fixed over the current step.

4.3. Notes and References

The augmented Lagrangian merit function was first proposed as an exact penalty function by Fletcher (1972). See also (Bertsekas 1982) and (Boggs and Tolle 1980). It was suggested as a merit function in (Powell and Yuan 1986) and in a slightly different form in (Boggs and Tolle 1984) and (Boggs and Tolle 1989). See (Byrd and Nocedal 1991) for its application to reduced Hessian methods. The Wolfe conditions have been studied by numerous authors; we recommend (Nocedal 1992) or (Dennis and Schnabel 1983).

Schittkowski (1981) and (Schittkowski 1983) developed the form for inequalities given by ϕ_{FI}. This form has been further developed and incorporated into a highly successful algorithm by Gill, Murray, Saunders and Wright (1986). (Boggs, Tolle and Kearsley 1991*b*) proposed the form given by ϕ_{FZ}.

The ℓ_1 exact penalty function was originally suggested as a merit function by Han (1977) where he obtained the first global convergence results for SQP methods. See (Fletcher 1981) for a good introduction to nondifferentiable optimization and the use of this function. See also (Polak 1989) and (Wolfe 1975). Byrd and Nocedal (1991) study the ℓ_1 merit function in the context of reduced Hessian methods. The analysis for the ℓ_1 merit function applies to the corresponding ℓ_p merit function with the only change ocurring in (4.20) where the ∞-norm and the 1-norm are replaced by the p-norm and the q-norm with $1/p + 1/q = 1$.

5. Global to Local Behavior

In the previous two sections we have examined conditions that imply that the basic SQP algorithm will converge. Section 3 dealt with local convergence where the emphasis was on the rates of convergence, while Section 4 was concerned with obtaining convergence from remote starting points; the implicit hope being that the two theories would come together to produce a unified theory that would be applicable to a given algorithm. For the global convergence theories, where the process has to be controlled by a merit function, it was seen in Section 4 that convergence of $\{\boldsymbol{x}^k\}$ to a critical point of ϕ is all that can be demonstrated. Assuming that this critical point is a solution of (NLP), the question that arises is whether or not the conditions for the local convergence theories are eventually satisfied by this sequence. If they are then the more rapid rates of local convergence can be achieved. In this section we discuss three questions concerning this possibility: will the correct active set be chosen by (QP) when $\boldsymbol{x}^k$ is close to $\boldsymbol{x}^*$; will B_k eventually approximate $H\mathcal{L}^*$ in one of the ways stipulated in Section 3 that yields rapid local convergence; and will the merit function allow a steplength of one if the underlying process can achieve rapid local convergence?

Recall that the local convergence theory was proven for equality-constrained problems only. This was based on the assumption that the active inequality constraints at the solution of (NLP) would be identified by (SQP) when $\boldsymbol{x}^k$ gets close to the solution. The question of when the active constraints for (QP) will be the same as those for (NLP) is resolved by using a perturbation theory result as follows. Consider (NLP) with only inequality constraints

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) \\ &\text{subject to: } \boldsymbol{g}(\boldsymbol{x}) \;\; \leq \;\; \mathbf{0} \end{aligned} \tag{5.1}$$

and the quadratic program with the same solution

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad \nabla f(\boldsymbol{x}^*)^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{x}^*) + \tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathrm{t}}B(\boldsymbol{x}-\boldsymbol{x}^*) \\ &\text{subject to: } \nabla \boldsymbol{g}(\boldsymbol{x}^*)^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{x}^*) + \boldsymbol{g}(\boldsymbol{x}^*) \;\; \leq \;\; \mathbf{0} \end{aligned} \tag{5.2}$$

where the only restriction on B is that

$$\boldsymbol{y}^{\mathrm{t}}B\boldsymbol{y} > 0 \tag{5.3}$$

for all $\boldsymbol{y}$ such that $\nabla \boldsymbol{g}(\boldsymbol{x}^*)^{\mathrm{t}}\boldsymbol{y} = 0$. It is easily verified that $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is a strong local minimizer of both (5.1) and (5.2). This implies that the active sets are the same and that strict complementary slackness (assumption **A3**) holds for both. The quadratic programming approximation to (5.1) is

$$\begin{aligned} \underset{\boldsymbol{d}_x}{\text{minimize}} \quad & \nabla f(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \tfrac{1}{2}\boldsymbol{d}_x{}^{\mathrm{t}}B_k\boldsymbol{d}_x \\ \text{subject to:} \ & \nabla \boldsymbol{g}(\boldsymbol{x}^k)^{\mathrm{t}}\boldsymbol{d}_x + \boldsymbol{g}(\boldsymbol{x}^k) \ \le \ 0. \end{aligned} \tag{5.4}$$

A standard perturbation argument now asserts that if $\boldsymbol{x}^k$ is close enough to $\boldsymbol{x}^*$ and B_k is close enough to B, then the active set for (5.4) is the same as the active set for (5.2). It follows that the active sets for (5.1) and (5.4) are the same. As a result, if $\boldsymbol{x}^k$ is sufficiently close to $\boldsymbol{x}^*$ and B_k is sufficiently close to any matrix B satisfying (5.3) then (5.4) will identify the correct active set. The condition (5.3) will be satisfied if the B_k are always positive definite or if they satisfy the condition

$$\lim_{k\to\infty} \mathcal{P}^k(B_k - H\mathcal{L}^*)\mathcal{P}^k = 0. \tag{5.5}$$

This latter condition implies two-step superlinear convergence by Theorem (3.5).

From Section 3 linear, superlinear, or two-step superlinear convergence is assured if B_k approaches $H\mathcal{L}^*$ in one of the ways hypothesized in Theorems 3.2–3.5. Since the initial matrix is unlikely in practice to be a good approximation to $H\mathcal{L}^*$, it is reasonable to ask under what conditions the subsequent B_k will be good enough approximations to $H\mathcal{L}^*$ for these results of Section 3 to hold. If the B_k satisfy the secant equation (3.19) then it is reasonable to expect that the B_k will converge to $H\mathcal{L}^*$ provided that the directions, $\boldsymbol{d}_x$, span $\mathcal{R}^n$ repeatedly, for example, if each set of n directions is (uniformly) linearly independent. Some recent research supports this expectation. However, if positive definiteness is maintained, it is not possible for the B_k to converge to $H\mathcal{L}^*$ unless the latter is at least positive semidefinite. So for the general case, further analysis is necessary. As seen in Section 3, (3.16) is equivalent to

$$\lim_{k\to\infty} \left\{ \frac{\mathcal{P}^*(B_k - H\mathcal{L}^*)\mathcal{P}^*\boldsymbol{s}_k}{\|\boldsymbol{s}_k\|} + \frac{\mathcal{P}^*(B_k - H\mathcal{L}^*)\mathcal{Q}^*\boldsymbol{s}_k}{\|\boldsymbol{s}_k\|} \right\} = 0, \tag{5.6}$$

where $\boldsymbol{s}_k = (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$. For superlinear convergence both of these terms must tend to zero if the directions $\boldsymbol{s}_k / \|\boldsymbol{s}_k\|$ repeatedly span $\mathcal{R}^n$. Thus for positive definite B_k superlinear convergence is not likely unless the second term goes to zero, which is the type of tangential convergence mentioned in Section 3.2. However, two step superlinear convergence, in which the first term of (5.6) goes to zero, is possible for positive definite updates and for reduced Hessian updates as well, provided that convergence occurs and steplengths of one are acceptable. Thus, it is important from the point of view of obtaining a rapid local rate of convergence that a steplength of one be taken near $\boldsymbol{x}^*$.

For the general algorithm, however, the use of the merit function with the line search continues throughout the course of the algorithm. Thus, the line search for the merit function should allow a steplength of one if the matrix approximations are such that rapid local convergence is possible. In the case of the augmented Lagrangian merit function it can be shown, using arguments similar to those in Section 4, that

$$\begin{aligned}\frac{\phi_F(\boldsymbol{x}^{k+1})-\phi_F(\boldsymbol{x}^k)}{\|\boldsymbol{d}_x\|^2} \quad \leq \quad & -\tfrac{1}{2}\beta_1 + \gamma_2\epsilon - \eta\gamma_3\epsilon^2 \\ & + \frac{\mathcal{P}^k[B_k - H\mathcal{L}(\boldsymbol{x}^k, \overline{\boldsymbol{u}}(\boldsymbol{x}^k))]\boldsymbol{d}_x}{\|\boldsymbol{d}_x\|} \\ & + \mathcal{O}(\|\boldsymbol{d}_x\|)\,.\end{aligned} \tag{5.7}$$

Then, if the process is converging superlinearly, the penultimate term in (5.7) tends to zero by Theorem 3.4 and the right hand side of (5.7) is ultimately less than zero, thus showing that a steplength of one decreases ϕ_F. A slight extension to this argument shows that the Wolfe condition (4.13) also can be satisfied for a steplength of one.

A significant disadvantage of the merit function ϕ_1 is that it may not allow a steplength of $\alpha = 1$ near the solution, no matter how good the approximation of (QP) to (NLP). This phenomenon, which can prohibit superlinear convergence, is called the Maratos effect. Several procedures have been proposed to overcome this problem, including so-called nonmonotone line searches, i.e., techniques that allow the merit function to increase over one or more iterations (see Section 7).

5.1. Notes and References

The perturbation argument showing conditions under which (QP) has the same active set as (NLP) is derived from the work of Robinson (1974) who proves a general perturbation result. For a discussion of the convergence of matrix updates see Boggs and Tolle (1994), Ge and Powell (1983) and Stoer (1984).

The Maratos effect, i.e., the fact that the ℓ_1 merit function may not allow a steplength of one even when the iterates are close to the solution and B_k is a good approximation to $H\mathcal{L}^*$, was discussed by Chamberlain, Lemarechal, Pedersen and Powell (1982). These authors suggested the "watchdog" technique, which is essentially a nonmonotone line search method. (See also (Bonnans et al. 1992).) The fact that the augmented Lagrangian merit function does not suffer from this problem has been shown by many authors.

6. SQP Trust Region Methods

Trust region algorithms have become a part of the arsenal for solving unconstrained optimization problems, so it is natural to attempt to extend the ideas to solving constrained optimization problems and, in particular, to SQP methods. In this section an outline of the basic ideas of this approach will be provided. As the subject is still in a state of flux, no attempt will be made to give a comprehensive account of the algorithms that have been proposed. The discussion will be limited to equality-constrained problems; the inclusion of inequality constraints into trust region algorithms has been the subject of little research.

A major difficulty associated with SQP methods arises from trying to ensure that (QP) has a solution. As was seen in the preceding, the requirement that the Hessian approximations be positive definite on the null space of the constraints is difficult to guarantee short of requiring the B_k to be positive definite. The trust region methods attempt to avoid this difficulty by the addition of a bounding constraint. Since this added constraint causes the feasible region to be bounded the subproblem is guaranteed to have a solution independent of the choice of B_k provided the feasible region is nonempty To be precise, a straightforward trust region adaptation of the SQP method would generate the iterates by solving the problem

$$\begin{array}{ll} \underset{\boldsymbol{d}_x}{\text{minimize}} & \nabla f(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \frac{1}{2} \boldsymbol{d}_x{}^{\mathrm{t}} B_k \, \boldsymbol{d}_x \\ \text{subject to:} & \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \boldsymbol{h}(\boldsymbol{x}^k) \;=\; \boldsymbol{0} \\ & \qquad\qquad \|S\boldsymbol{d}_x\|^2 \;\leq\; \triangle_k^2 \end{array} \tag{6.1}$$

where S is a positive diagonal scaling matrix and $\triangle_k$ is the *trust region radius*. In this discussion we assume the norm is the Euclidean norm but other norms have been used in the trust region constraint. The trust region radius is updated at each iteration based on a comparison of the actual decrease in a merit function to the decrease predicted by the model. If there is good agreement the radius is maintained or increased; if not, it is decreased.

It is worth noting that the necessary condition for $\boldsymbol{d}_x$ to be a solution to (6.1) is that

$$(B_k + \mu\, S^2)\boldsymbol{d}_x = -\nabla \boldsymbol{h}(\boldsymbol{x}^k)\boldsymbol{u} - \nabla f(\boldsymbol{x}^k) \tag{6.2}$$

for some multiplier vector $\boldsymbol{u}$ and some nonnegative scalar μ. This equation is used to generate approximate solutions to (6.1).

The removal of the positive definite requirement on B_k does not come without cost. The additional constraint is a quadratic inequality constraint and hence it is not a trivial matter to find a good approximate solution to (6.1). Moreover, there is the bothersome possibility that the solution sets of the linear constraints and the trust region constraint may be disjoint. For this reason, research on trust region methods has centered on finding different types of subproblems that have feasible solutions but still capture the essence of the quadratic subproblem. Several avenues of investigation are summarized below.

One approach is to *relax* the linear constraints in such a way that the resulting problem is feasible. In this case the linear constraint in (6.1) is replaced by

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \theta_k \boldsymbol{h}(\boldsymbol{x}^k) = \mathbf{0} \tag{6.3}$$

where $0 \leq \theta_k \leq 1$. A major difficulty with this approach is the problem of choosing θ_k so that (6.3) together with the trust region constraint has a solution.

A second approach is to replace the equality constraints by a least squares approximation. Then the equality constraints in (6.1) become the quadratic constraint

$$\left\| \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \boldsymbol{h}(\boldsymbol{x}^k) \right\|^2 \leq (\rho_k)^2 \tag{6.4}$$

where ρ_k is an appropriate value. One choice of ρ_k is the error in the linear constraints evaluated at the "Cauchy" point. The Cauchy point is defined to be the optimal step in the steepest descent direction for the function

$$\left\| \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \boldsymbol{h}(\boldsymbol{x}^k) \right\|^2$$

i.e., the step, s_{cp}, which minimizes this function in the steepest descent direction. This yields

$$(\rho_k)^2 = \left\| \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \mathrm{s}_{cp} + \boldsymbol{h}(\boldsymbol{x}^k) \right\|^2 .$$

Another possibility is to take ρ_k to be any value of $\left\| \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \mathrm{s} + \boldsymbol{h}(\boldsymbol{x}^k) \right\|$ for which

$$\sigma_1 \triangle_k \leq \|\mathrm{s}\| \leq \sigma_2 \triangle_k$$

where $0 < \sigma_1 \leq \sigma_2 < 1$. This approach assures that the quadratic subproblem is always feasible, but at the cost of some extra computation to obtain ρ_k. Moreover, while the resulting subproblem has only two quadratic constraints, finding a good approximate solution quickly is a matter that has not been completely resolved.

Finally, in a Reduced Hessian approach to the trust region method the linear constraint in (6.1) is replaced by

$$\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \boldsymbol{d}_x + \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \mathrm{s}_k = \mathbf{0} \tag{6.5}$$

where s_k is the solution of

$$\begin{array}{ll} \underset{\mathrm{s}}{\text{minimize}} & \left\| \nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}} \mathrm{s} + \boldsymbol{h}(\boldsymbol{x}^k) \right\|^2 \\ \text{subject to:} & \|\mathrm{s}\| \ \leq \ \tau \triangle_k \end{array}$$

for $\tau \in (0,1)$. Using the decomposition (3.27) from Section 3.4

$$d_x = Z_k\,\mathbf{p}_Z + Y_k\,\mathbf{p}_Y$$

where the columns of $Z(\boldsymbol{x}^k)$ are a basis for the null space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)^{\mathrm{t}}$ and the columns of Y_k are a basis for the range space of $\nabla \boldsymbol{h}(\boldsymbol{x}^k)$, it follows from (6.5) that

$$Y_k\,\mathbf{p}_Y = -\mathbf{s}_k.$$

As in Section 3.4, the null space component, $\mathbf{p}_Z$, can now be seen to be the solution of the quadratic problem

$$\begin{array}{ll} \underset{\mathbf{p}_Z}{\text{minimize}} & \left[f(\boldsymbol{x}^k) + B_k\mathbf{s}_k\right]^{\mathrm{t}} \mathbf{p}_Z + \frac{1}{2}\mathbf{p}_Z{}^{\mathrm{t}} Z_k{}^{\mathrm{t}} B_k\, Z_k\,\mathbf{p}_Z \\ \text{subject to:} & \left\|\mathbf{p}_Z\right\|^2 \;\leq\; (\triangle_k)^2 - \left\|\mathbf{s}_k\right\|^2. \end{array}$$

A great deal of research has been done on the subject of minimizing a quadratic function subject to a trust region constraint so that quick and accurate methods for approximating the solutions to these two problems are available.

Much of the work to be done in transforming these approaches into algorithms for which local and global convergence theorems can be proven is similar in nature to that which must be done for standard SQP methods. In particular, a method for updating the matrices B_k needs to be specified (a wider class of updating schemes is now available, at least in theory, because there is no necessity of requiring them to have the positive definiteness properties) and a merit function has to be specified. As was shown in Section 3, local superlinear convergence depends on the steps approaching the Newton-SQP steps as the solution is neared. In particular, this requires that the modified constraints reduce to the linearized constraint of (ECQP) and that the trust region constraint not be active so that steplengths of one can be accepted. Conditions under which these events will occur have not been established. Global convergence theorems have been proven for most of the above variations of the trust region method by utilizing either the ℓ_1 or the augmented Lagrangian merit function under the assumptions **C1** and **C2**.

6.1. Notes and References

An introductory exposition of trust region methods for unconstrained optimization can be found in the book by Dennis and Schnabel (1983). Methods for minimizing a quadratic function subject to a trust region constraint can be found in (Gay 1981) and (Moré and Sorensen 1983).

The first application of trust region methods to the constrained problem appears to be that of Vardi (1985), who used the constraint (6.3) in place of the linearized equality constraints. This approach was also used by Byrd, Schnabel and Schultz (1985). The introduction of the quadratic constraint (6.4) as a substitute for the linear constraint is due to Celis, Dennis and Tapia (1985), who used the error at the "Cauchy point" as the ρ_k. A global convergence theory for this strategy is given in (El-Alem 1991). Powell and Yuan (1986) suggested the second version of this approach mentioned in the text. Yuan (1990) proposed solution techniques for the resulting subproblem. The reduced Hessian approach has been introduced by Omojokun (1989) who also considers the case when inequality constraints are present. This approach has also been used by (Lalee, Nocedal and Plantega 1993) for large-scale problems.

7. Practical Considerations

Our goal throughout this survey has been to concentrate on those theoretical properties that bear on actual implementations, but we have mentioned some of the difficulties that arise in the implementation of an SQP algorithm when the assumptions that we have made are not satisfied. In this section we elaborate on a few of the more important of these difficulties and suggest some common computational techniques that can be used to work around them. No attempt is made to be complete; the object is to give an indication of the issues involved. We also discuss briefly the extension of the SQP algorithm to the large scale case, where special considerations are necessary to create an efficient algorithm.

7.1. The Solution of (QP)

The assumptions behind the theory in Sections 3 and 4 imply that (QP) always has a solution. In practice, of course, this may not be the case. (QP) may be infeasible or the objective function may be unbounded on the feasible set and have no local minima. As stated earlier, surveying methods for solving (QP) is beyond the scope of this paper, but we discuss some ways of continuing the computations when these difficulties arise.

One technique to avoid infeasibilities is to relax the constraints by using the trust region methods in Section 6. Another approach is to take $\boldsymbol{d_x}$ to be some convenient direction when (QP) is infeasible, e.g., the steepest descent direction for the merit function. The infeasibility of (QP) is usually detected during a "Phase I" procedure that is used to obtain a feasible point with which to begin the quadratic programming algorithm. If the constraints are inconsistent, then there are no feasible points and the Phase I procedure will fail. However, the direction, $\boldsymbol{d_x}$, obtained in this case can sometimes be used to produce directions that reduce the constraint infeasibilities and can thus be used to improve $\boldsymbol{x}^k$. For example, the phase I procedure known as the "big M" method modifies (QP) by adding one new variable, say θ, in such a way that the new problem

$$\begin{array}{lrcl}\underset{d_x}{\text{minimize}} & \nabla f(x^k)^{\mathrm{t}} d_x + \frac{1}{2} d_x{}^{\mathrm{t}} B_k d_x + M\theta & & \\ \text{subject to:} & \nabla h(x^k)^{\mathrm{t}} d_x + h(x^k) & = & 0 \\ & \nabla g(x^k)^{\mathrm{t}} d_x + g(x^k) - \theta \mathbf{e} & \leq & 0 \end{array} \tag{7.1}$$

is feasible. In this form M is a constant and $\mathbf{e}$ is the vector of ones. For $\theta^0 \equiv \max\{g_i(x^k) \ : \ g_i(x^k) > 0\}$ the initial point $(d_x, \theta) = (\mathbf{0}, \theta^0)$ is feasible for (7.1). The constant M is chosen large enough so that, as (7.1) is being solved, θ is forced to be reduced. Once $\theta \leq 0$, d_x is a feasible point for (QP). If θ cannot be reduced to zero, then the inequalities are inconsistent. Nevertheless, it can be shown under certain conditions that the resulting d_x is a descent direction for the merit function ϕ_F.

If the problem is unbounded, but feasible, then the difficulty is due to the structure of the Hessian approximation, B_k. For example, if B_k is not positive definite then a multiple of the identity (or non-negative diagonal matrix) can be added to B_k to ensure that it is positive definite. Note that adding a strictly positive diagonal matrix to B_k is equivalent to using a trust region constraint (see (6.2)).

Finally, in cases where (QP) does not yield approximate multipliers for nonlinear program such as when (QP) is infeasible or the matrix $G(x^k)$ has linearly dependent columns, any reasonable approximation of the multipliers will usually suffice. The most common approach in these cases is to use a least squares multiplier approximation. If (NLP) does not satisfy **A2** then at best the theoretical convergence rate will be slowed (if x^* is a simple degeneracy) and at worst even the Newton method may fail.

7.2. Adjusting the Merit Function Parameter

It was shown in Section 4 that setting the parameter ρ to ensure that the direction d_x is a descent direction for the merit function ϕ_1 given by (4.18) can be relatively straightforward. Setting the parameter for ϕ_F is only a little more difficult. In either case adjusting the parameter can cause both theoretical and computational difficulties. In theory, there is no problem with having the parameter large. Indeed, if only increases in the parameter are allowed, the assumptions of Section 4 coupled with a reasonable adjustment strategy will lead to a provably finite value of the parameter and convergence can be proved using the techniques of Section 4. Computationally, however, having a large value of the parameter implies that there will be too much emphasis on satisfying the constraints. The result will be that the iterates will be forced to follow the constraint boundary closely which, in highly nonlinear problems, can cause the algorithm to be slow. A strategy for only allowing increases in the parameter can lead to an excessively large value due entirely to an early iterate's being far from the solution.

Ideally the parameter should be adjusted up or down at various stages of the iteration process to ensure both good theoretical and computational performance. Some strategies for this have been proposed. For example, it is possible to allow controlled decreases in the parameter that ensure that the predicted decrease in the merit function is not dominated by a decrease in the constraint infeasibilities. For such choices it is still possible to prove convergence.

7.3. Nonmonotone Decrease of the Merit Function

Global convergence results are usually proved by insisting that an appropriate merit function be sufficiently reduced at each iteration. Sometimes, however, it is more efficient computationally to be less conservative and to allow steps to be accepted even if the merit function is temporarily increased. If, for example, the merit function is forced to be reduced over any fixed number of iterations, then convergence follows. In practice such strategies have been quite successful, especially near the solution. As a particular example, note that the merit function ϕ_1 may not allow a steplength of one near the solution. One remedy for this is to accept the full step temporarily even if ϕ_1 increases. Then, if ϕ_1 is not sufficiently reduced after a small number of steps, the last good iterate is restored and reduction in ϕ_1 is required for the next iteration.

As a second example, it was noted in Section 4 that the use of ϕ_F requires the evaluation of f, $\boldsymbol{h}$, $\boldsymbol{g}$, and their gradients just to test a prospective point for acceptance. This difficulty can be circumvented by using an "approximate" or "working" merit function that only requires evaluation of f, $\boldsymbol{h}$, and $\boldsymbol{g}$, and no gradients, to test a point. For example, ϕ_F could be approximated (in the equality-constrained case) by

$$\phi_F^k(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{t}}\overline{\boldsymbol{u}}(\boldsymbol{x}^k) + \tfrac{1}{2}\eta \, \|\boldsymbol{h}(\boldsymbol{x})\|^2$$

where $\overline{\boldsymbol{u}}(\boldsymbol{x}^k)$ stays fixed throughout the k^{th} iteration. This is then coupled with a strategy to monitor the iterations to ensure that ϕ_F is sufficiently reduced after a certain number of iterations as discussed above.

7.4. Large Scale Problems

Efficient SQP algorithms in the large scale case depend on carefully addressing many factors. In this section we mention two of these, namely problem structure and the solution of large scale quadratic programs.

Problems are considered large if, to solve them efficiently, either their structure must be exploited or the storage and manipulation of the matrices involved must be handled in a special manner. The most obvious structure, and the one most commonly considered, is sparsity of the matrices $\nabla\boldsymbol{h}$, $\nabla\boldsymbol{g}$, and $H\mathcal{L}$. Typically in large scale problems most constraints depend only on a few of the variables and the objective function is " partially separable" (a partially separable function is, for example made up of a sum of functions, each of which depends only on a few of the variables). In such cases the matrices are sparse. In other cases, one or more of the matrices is of low rank, and this structure can also be exploited.

A key to using SQP in the large scale case is to be able to solve, or to solve approximately, the large quadratic programming subproblems. In the small scale case, it rarely matters how (QP) is solved and it usually makes sense to solve it completely. Depending on the algorithm used in the large scale case, it is often essential to realize that (QP) at iteration $k+1$ differs only slightly from that at iteration k. In these cases, the active set may remain the same, or change only slightly, and it follows that solving (approximately) the next (QP) can be accomplished quickly. It must be shown, however, that approximate solutions of (QP) are descent directions for an appropriate merit function.

Recently, efficient interior point methods have been developed for solving large scale linear programs; these ideas are now being applied to large scale quadratic programs. One such method is the "subspace" method where, at each iteration, a low-dimensional subspace is chosen and (QP) restricted to that subspace is solved. A step in the resulting direction is calculated and the procedure iterated. It has been shown that any number of iterations of this technique gives rise to a descent direction for an augmented Lagrangian merit function; an SQP algorithm based on this has shown promise.

7.5. Notes and References

See (Fletcher 1981) for an introduction to the solution of quadratic programs using classical techniques, including the Big M method. Recently there have been numerous papers on the application of interior-point methods to quadratic programming problems. This is still the subject of intense research and there are no good survey papers on the subject. See (Vanderbei and Carpenter 1993) and (Boggs, Domich, Rogers and Witzgall 1991*a*) or (Boggs, Domich, Rogers and Witzgall 1994*a*) for a discussion of two different approaches.

Adjusting the penalty parameter in the augmented Lagrangian merit function is discussed in (Schittkowski 1981), (Schittkowski 1983), (Powell and Yuan 1986), (Byrd and Nocedal 1991), (Boggs, Tolle and Kearsley 1994*b*), (Byrd and Nocedal 1991), and (El-Alem 1992). The idea of allowing the merit function to increase temporarily is an old one. It is sometimes referred to as a nonmonotone line search; for a general discussion see (Grippo, Lampariello and Lucidi 1986). Its use with the ℓ_1 merit function is discussed in (Chamberlain et al. 1982). Using approximate merit functions is suggested in (Powell and Yuan 1986), (Boggs and Tolle 1989), and (Boggs et al. 1994*b*).

Based on the work to date there is significant promise for SQP in the large scale case, but more work is required to compare SQP with other large scale techniques. For work to date see (Murray 1994), (Murray and Prieto to appear), and (Boggs et al. 1994*b*).

Acknowledgments: The authors would like to thank Ubaldo Garcia-Palomares, Anthony Kearsley, Stephen Nash, Dianne O'Leary, and G. W. Stewart for reading and commenting on early drafts of the paper.

REFERENCES

D. P. Bertsekas (1980), Variable metric methods for constrained optimization based on differentiable exact penalty functions, in *Proceedings of the Eighteenth Allerton Conference on Communications, Control and Computing*, Allerton Park, Illinois, pp. 584–593.

D. P. Bertsekas (1982), *Constrained Optimization and Lagrange Multipliers Methods*, Academic Press, London.

P. T. Boggs and J. W. Tolle (1980), 'Augmented Lagrangians which are quadratic in the multiplier', *Journal of Optimization Theory and Applications* **31**, 17–26.

P. T. Boggs and J. W. Tolle (1984), 'A family of descent functions for constrained optimization', *SIAM Journal on Numerical Analysis* **21**, 1146–1161.

P. T. Boggs and J. W. Tolle (1989), 'A strategy for global convergence in a sequential quadratic programming algorithm', *SIAM Journal on Numerical Analysis* **21**, 600–623.

P. T. Boggs and J. W. Tolle (1994), 'Convergence properties of a class of rank-two updates', *SIAM Journal on Optimization* 4, 262–287.

P. T. Boggs, P. D. Domich, J. E. Rogers and C. Witzgall (1991*a*), 'An interior point method for linear and quadratic programming problems', *Mathematical Programming Society Committee on Algorithms Newsletter* (19), 32–40.

P. T. Boggs, P. D. Domich, J. E. Rogers and C. Witzgall (1994*a*), An interior-point method for general large scale quadratic programming problems, Internal Report (in process), National Institute of Standards and Technology.

P. T. Boggs, J. W. Tolle and A. J. Kearsley (1991*b*), A merit function for inequality constrained nonlinear programming problems, Internal Report 4702, National Institute of Standards and Technology.

P. T. Boggs, J. W. Tolle and A. J. Kearsley (1994*b*), A practical algorithm for general large scale nonlinear optimization problems, Internal report, National Institute of Standards and Technology.

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.

J. F. Bonnans, E. R. Panier, A. L. Tits and J. L. Zhou (1992), 'Avoiding the Maratos effect by means of a nonmonotone line search II. Inequality constrained problems—feasible iterates', *SIAM Journal on Numerical Analysis* **29**, 1187–1202.

C. G. Broyden, J. E. Dennis, Jr. and J. J. Moré (1973), 'On the local and superlinear convergence of quasi-Newton methods', *Journal of the Institute of Mathematical Applications* **12**, 223–246.

R. H. Byrd and J. Nocedal (1991), 'An analysis of reduced Hessian methods for constrained optimization', *Mathematical Programming* **49**, 285–323.

R. H. Byrd and R. B. Schnabel (1986), 'Continuity of the null space basis and constrained optimization', *Mathematical Programming* **35**, 32–41.

R. H. Byrd, R. B. Schnabel and G. A. Schultz (1985), 'A trust-region strategy for nonlinearly constrained optimization', *SIAM Journal on Numerical Analysis* **24**, 1152–1170.

R. H. Byrd, R. A. Tapia and Y. Zhang (1992), 'An SQP augmented Lagrangian BFGS algorithm for constrained optimization', *SIAM Journal on Optimization* **2**, 210–241.

M. R. Celis, J. E. Dennis, Jr. and R. A. Tapia (1985), A trust-region strategy for nonlinear equality constrained optimization, in *Numerical Optimization 1984* (P. Boggs, R. Byrd and R. Schnabel, eds), SIAM, Philadelphia, pp. 71–82.

R. Chamberlain, C. Lemarechal, H. C. Pedersen and M. J. D. Powell (1982), 'The watchdog technique for forcing convergence in algorithms for constrained optimization', *Mathematical Programming Study* **16**, 1–17.

T. F. Coleman (1990), On characterizations of superlinear convergence for constrained optimization, in *Lectures in Applied Mathematics*, American Mathematical Society, Providence, Rhode Island, pp. 113–133.

T. F. Coleman and A. R. Conn (1984), 'On the local convergence of a quasi-Newton method for the nonlinear programming problem', *SIAM Journal on Numerical Analysis* **21**, 755–769.

T. F. Coleman and P. A. Feynes (1992), 'Partitioned quasi-Newton methods for nonlinear equality constrained optimization', *Mathematical Programming* **53**, 17–44.

T. F. Coleman and D. C. Sorensen (1984), 'A note on the computation of an orthonormal basis for the null space of a matrix', *Mathematical Programming* **29**, 234–242.

J. E. Dennis, Jr. and J. J. Moré (1977), 'Quasi-Newton methods, motivation and theory', *SIAM Review* **19**, 46–89.

J. E. Dennis, Jr. and R. B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey.

M. M. El-Alem (1991), 'A global convergence theory for the Celis-Dennis-Tapia trust region algorithm for constrained optimization', *SIAM Journal on Numerical Analysis* **28**, 266–290.

M. M. El-Alem (1992), A robust trust region algorithm with nonmonotonic penalty parameter scheme for constrained optimization, Department of Mathematical Sciences 92-30, Rice University.

R. Fletcher (1972), A class of methods for nonlinear programming, iii: Rates of convergence, in *Numerical Methods for Nonlinear Optimization* (F. A. Lootsma, ed.), Academic Press, New York, pp. 371–382.

R. Fletcher (1981), *Practical Methods of Optimization: Volume 2*, John Wiley & Sons, New York.

R. Fontecilla (1988), 'Local convergence of secant methods for nonlinear constrained optimization', *SIAM Journal on Numerical Analysis* **25**, 692–712.

R. Fontecilla, T. Steihaug and R. A. Tapia (1987), 'A convergence theory for a class of quasi-Newton methods for constrained optimization', *SIAM Journal on Numerical Analysis* **24**, 1133–1151.

M. Fukushima (1986), 'A successive quadratic programming algorithm with global and superlinear convergence properties', *Mathematical Programming* **35**, 253–264.

D. Gabay (1982), 'Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization', *Mathematical Programming Study* **16**, 18–44.

U. M. Garcia-Palomares and O. L. Mangasarian (1976), 'Superlinearly convergent quasi-Newton methods for nonlinearly constrained optimization problems', *Mathematical Programming* **11**, 1–13.

D. M. Gay (1981), 'Computing optimal locally constrained steps', *SIAM Journal on Scientific and Statistical Computing* **2**, 186–197.

R. Ge and M. J. D. Powell (1983), 'The convergence of variable metric matrices in unconstrained optimization', *Mathematical Programming* **27**, 123–143.

J. C. Gilbert (1993), Superlinear convergence of a reduced BFGS method with piecewise line-search and update criterion, Rapport de Recherche 2140, Institut National de Recherche en Informatique et en Automatique.

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

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, Technical Report SOL 2, Department of Operations Research, Stanford University.

P. E. Gill, W. Murray, M. A. Saunders, G. W. Stewart and M. H. Wright (1985), 'Properties of a representation of a basis for the null space', *Mathematical Programming* **33**, 172–186.

S. T. Glad (1979), 'Properties of updating methods for the multipliers in augmented Lagrangians', *Journal of Optimization Theory and Applications* **28**, 135–156.

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

L. Grippo, F. Lampariello and S. Lucidi (1986), 'A nonmonotone line search technique for Newton's method', *SIAM Journal on Numerical Analysis* **23**, 707–716.

S.-P. Han (1976), 'Superlinearly convergent variable metric algorithms for general nonlinear programming problems', *Mathematical Programming* **11**, 263–282.

S.-P. Han (1977), 'A globally convergent method for nonlinear programming', *Journal of Optimization Theory and Applications* **22**, 297–309.

M. Lalee, J. Nocedal and T. Plantega (1993), On the implementation of an algorithm for large-scale equality optimization, Department of Electrical Engineering and Computer Science NAM 09-93, Northwestern University.

D. G. Luenberger (1984), *Linear and Nonlinear Programming, 2nd Edition*, Addison-Wesley, Reading, Massachusetts.

J. Moré and D. C. Sorensen (1983), 'Computing a trust region step', *SIAM Journal of Scientific and Statistical Computing* **4**, 553–572.

W. Murray (1994), Algorithms for large nonlinear problems, in *Mathematical Programming: State of the Art 1994* (J. R. Birge and K. G. Murty, eds), University of Michigan, Ann Arbor, MI, pp. 172–185.

W. Murray and F. J. Prieto (to appear), 'A sequential quadratic programming algorithm using an incomplete solution of the subproblem', *SIAM Journal on Optimization.*

S. G. Nash and A. Sofer (1995), *Linear and Nonlinear Programming*, McGraw-Hill, New York.

J. Nocedal (1992), 'Theory of algorithms for unconstrained optimization', *Acta Numerica* **1991**, 199–242.

J. Nocedal and M. L. Overton (1985), 'Projected Hessian updating algorithms for nonlinearly constrained optimization problems', *SIAM Journal on Numerical Analysis* **22**, 821–850.

E. M. Omojokun (1989), Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints, PhD thesis, University of Colorado.

J. M. Ortega and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations In Several Variables*, Academic Press, New York.

E. R. Panier and A. L. Tits (1993), 'On combining feasibility, descent and superlinear convergence in inequality constrained optimization', *Mathematical Programming* **59**, 261–276.

E. Polak (1989), Basics of minimax algorithms, in *Nonsmooth Optimization and Related Topics* (F. H. Clark, V. F. Dem'yanov and F. Giannessi, eds), Plenum, New York, pp. 343–369.

M. J. D. Powell (1977), A fast algorithm for nonlinearly constrained optimization calculations, in *Numerical Analysis Dundee, 1977* (G. A. Watson, ed.), Springer-Verlag, Berlin, pp. 144–157.

M. J. D. Powell (1978*a*), 'Algorithms for nonlinear constraints that use Lagrangian functions', *Mathematical Programming* **14**, 224–248.

M. J. D. Powell (1978*b*), The convergence of variable metric methods for nonlinearly constrained optimization calculations, in *Nonlinear Programming 3* (O. Mangasarian, R. Meyer and S. Robinson, eds), Academic Press, New York, NY, pp. 27–64.

M. J. D. Powell and Y. Yuan (1986), 'A recursive quadratic programming algorithm that uses differentiable exact penalty functions', *Mathematical Programming* **35**, 265–278.

S. M. Robinson (1974), 'Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms', *Mathematical Programming* **7**, 1–16.

K. Schittkowski (1981), 'The nonlinear programming method of Wilson, Han, and Powell with an augemented Lagrangian type line search function, part 1: Convergence analysis', *Numerische Mathematik* **38**, 83–114.

K. Schittkowski (1983), 'On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function', *Math. Operationsforsch. U. Statist., Ser. Optimization* **14**, 197–216.

J. Stoer (1984), 'The convergence of matrices generated by rank-2 methods from the restricted β-class of Broyden', *Numerische Mathematik* **44**, 37–52.

R. A. Tapia (1974), 'A stable approach to Newton's method for optimization problems with equality constraints', *Journal of Optimization Theory and Applications* **14**, 453–476.

R. A. Tapia (1977), 'Diagonalized multiplier methods and quasi-Newton methods for constrained optimization', *Journal of Optimization Theory and Applications* **22**, 135–194.

R. A. Tapia (1978), Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation, in *Nonlinear*

Programming 3 (O. Mangasarian, R. Meyer and S. Robinson, eds), Academic Press, New York, NY, pp. 125–164.

R. A. Tapia (1988), 'On secant updates for use in general constrained optimization', *Mathematics of Computation* **51**, 181–202.

R. J. Vanderbei and T. J. Carpenter (1993), 'Symmetric indefinite systems for interior point methods', *Mathematical Programming* **58**, 1–32.

A. Vardi (1985), 'A trust-region algorithm for equality constrained minimization and implementation', *SIAM Journal on Numerical Analysis* **22**, 575–591.

R. B. Wilson (1963), A Simplicial Algorithm for Concave Programming, PhD thesis, Harvard University.

P. Wolfe (1975), A method of conjugate subgradients for minimizing nondifferentiable functions, in *Mathematical Programming Study 3* (M. Balinski and P. Wolfe, eds), North Holland, pp. 145–173.

Y. Yuan (1985), 'An only 2-step q-superlinear convergence example for some algorithms that use reduced Hessian approximations', *Mathematical Programming* **32**, 224–231.

Y. Yuan (1990), 'On a subproblem of trust region algorithms for constrained optimization', *Mathematical Programming* **47**, 53–63.