Chapter 3

NONLINEAR CONSTRAINTS

Even one nonlinear constraint considerably increases the difficulty of solving an optimization problem. It often pays to try and eliminate nonlinear constraints if at all possible. A measure of the increase in difficulty may be gauge from the problem of minimizing a quadratic function subject to one linear equality constraint, which may be solved by solving a single system of linear equations. If instead we have a quadratic constraint then we have the generalized eigenvalue problem. Such a problem may be solved only by iteration. An issue not present in linearly constrained problems is that of feasibility. It is in part the fact that simply to obtain a feasible point or to maintain feasibility is in general an infinite process that makes solving nonlinearly constrained problems hard.

As in the linearly constrained case we again consider first the equality-constrained problem.

3.1. Equality Constraints

The nonlinear equality-constrained problem may be expressed as follows:

$$\text{NEP} \qquad \begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) = 0, \end{array} \tag{3.1.1}$$

where $c(x)$ is an m-vector of nonlinear functions with i-th component $c_i(x)$, $i = 1,\ldots,m$, and f and $\{c_i\}$ are twice-continuously differentiable. Let $g(x)$ denote the gradient vector of $f(x)$, $a_i(x)$ the gradient vector of $c_i(x)$, and $A(x)$ the $m \times n$ Jacobian matrix of $c(x)$. A solution of NEP will be denoted by x^*.

The most powerful algorithms for solving NEP are based on seeking a point satisfying conditions that hold at the solution. The key role of optimality conditions in algorithm design has already been discussed in the context of *unconstrained* problems, in which the gradient of the objective function must vanish at the solution, and the Hessian matrix must be positive semidefinite. We seek analogous conditions for nonlinearly constrained problems.

Definition 3.1.1. *A point $\widehat{x} \in \mathbb{R}^n$ is* feasible *with respect to the constraint $c_i(x) = 0$ if $c_i(\widehat{x}) = 0$ (in which case we say that the constraint is* satisfied *at $\widehat{x}$). Otherwise,*

$\widehat{x}$ is infeasible*, and we say that the constraint is* violated *at $\widehat{x}$. A point $\widehat{x}$ is feasible with respect to the* set *of constraints $c(x) = 0$ if it is feasible with respect to every constraint.*

Definition 3.1.2. *The point x^* is a* local minimizer *(or local solution) of NEP if:*

1. *x^* is feasible with respect to all the constraints;*
2. *there exists a neighborhood $N(x^*)$ such that*

$$f(x^*) \leq f(x) \quad \text{for all feasible } x \in N(x^*). \tag{3.1.2}$$

If inequality (3.1.2) is strict for all feasible $x \in N(x^*)$, $x \neq x^*$, then x^* is said to be a *strong* or *strict* local minimizer. Otherwise, x^* is called a *weak* local minimizer.

In order to verify that Definition 3.1.2 applies at a feasible point, a characterization is needed of neighboring feasible points. In general, feasibility can be retained with respect to nonlinear equality constraints only by movement along a *nonlinear* path in $\mathbb{R}^n$, which is called a *feasible arc*. For example, the nonlinear constraint $x_1^2 + x_2^2 = 1$ defines a unit circle in $\mathbb{R}^2$, centered at the origin, whose curved boundary is a feasible arc. In general, a *feasible arc* is a directed differentiable curve $x(\alpha)$ emanating from a feasible point x, parameterized by the scalar α such that $x(0) = x$ and

$$c(x(\alpha)) = 0,$$

for all α satisfying $0 \leq \alpha \leq \bar{\alpha}$, where $\bar{\alpha} > 0$, and $dx(0)/d\alpha \neq 0$.

If a feasible arc exists at the feasible point x, then every neighborhood of x contains feasible points. To test optimality, a condition is needed that indicates whether a feasible arc exists. The Taylor series expansion of c along any direction p is

$$c(x + \epsilon p) = c(x) + \epsilon A(x)p + O(\epsilon^2 \|p\|^2), \tag{3.1.3}$$

where ϵ is any scalar. The points along a path from x to a neighboring feasible point will be feasible only if the direction of movement from x is *tangent* to a feasible arc. The relationship (3.1.3) implies that p will be tangent to a feasible arc only if

$$A(x)p = 0. \tag{3.1.4}$$

(If (3.1.4) does not hold, even an infinitesimal move along any path tangent to p will lead to a constraint violation.) Unfortunately, the nonlinearity of $c(x)$ means that condition (3.1.4) is not *sufficient* to ensure that p is tangent to a feasible arc. Requirements that $c(x)$ must satisfy in order to permit analysis of feasible arcs are called *constraint qualifications*, and have been widely studied. We consider only one constraint qualification for equality constraints. (For further discussion, see, e.g., [Kuhn and Tucker 1951], [Fiacco and McCormick 1968].)

Definition 3.1.3. *(Constraint qualification for equality constraints.) The constraint qualification with respect to the equality constraints $c(x) = 0$ holds at the feasible point x if every nonzero vector p satisfying (3.1.4) is tangent to a twice-differentiable feasible arc emanating from x.*

When the constraints are *linear*, the constraint qualification always holds. For nonlinear constraints, however, Definition 3.1.3 has the unsatisfactory feature that it cannot easily be checked. In contrast, the following result provides a computationally practical test for verifying satisfaction of the constraint qualification when the constraints are nonlinear. (See [Fiacco and McCormick 1968], for a proof.)

Theorem 3.1.1. *(Regular point for equality constraints.) If x is feasible and $A(x)$ has full rank, the constraint qualification holds at x, and x is said to be a regular point.* ∎

At a regular point, (3.1.4) provides a complete characterization of tangents to feasible arcs. Relation (3.1.4) states that p lies in the *null space* of the matrix $A(x)$. Given a point x, let $Z(x)$ denote a (non-unique) matrix whose columns form a basis for the null space of $A(x)$, so that

$$A(x)Z(x) = 0. \tag{3.1.5}$$

The relationship (3.1.5) is crucial in all subsequent analysis. (We shall henceforth use Z as a generic notation for a basis for the null space, where the relevant point x and matrix A should always be clear from context. Note that a basis for the null space exists even when $A(x)$ is rank-deficient.) By definition of a basis, every vector p such that $Ap = 0$ may be written as a linear combination of the columns of Z, so that

$$Ap = 0 \quad \text{if and only if} \quad p = Zp_Z$$

for some vector p_Z.

We now state the well known *first-order Karush-Kuhn-Tucker conditions* for a solution of NEP.

Theorem 3.1.2. *(First-order necessary optimality conditions.) If the constraint qualification holds at x^*, a necessary condition for x^* to be a local minimizer of NEP is that*

$$g(x^*) = A(x^*)^T\lambda^* = \sum_{i=1}^{m} \lambda_i^* a_i(x^*), \tag{3.1.6}$$

for some λ^*. Equivalently, if $Z(x)$ denotes a null-space basis for $A(x)$, then

$$Z(x^*)^T g(x^*) = 0. \tag{3.1.7}$$

Proof. Assume that x^* is a local minimizer of NEP, and consider all nonzero vectors p such that

$$A(x^*)p = 0. \tag{3.1.8}$$

(If there are none, then (3.1.6) must hold, since the columns of $A(x^*)$ span all of $\mathbb{R}^n$.)

By hypothesis, the constraint qualification holds at x^*, and hence any vector p satisfying (3.1.8) is tangent to a feasible arc emanating from x^*. From Definition 3.1.2, x^* can be a local minimizer only if f does not strictly decrease along any

such arc, which implies that $g(x^*)^T p = 0$. It follows from standard linear algebra that if $g(x^*)^T p = 0$ for *every* p satisfying (3.1.8), then $g(x^*)$ must lie entirely in the range space of $A(x^*)$, which gives the desired result (3.1.6). ∎

We emphasize the key role of the constraint qualification in proving this theorem. Without it, relationship (3.1.8) does not imply that p is tangent to a feasible arc, and hence there can exist directions satisfying (3.1.8) along which no feasible points exist. For example, the origin is the *only* feasible point in $\mathbb{R}^1$ with respect to the nonlinear constraint $x_1^2 = 0$, and hence no feasible arcs exist. Nonetheless, (3.1.8) is satisfied for every nonzero p because the Jacobian matrix vanishes at the origin.

The m-vector λ^* such that $g(x^*) = A(x^*)^T \lambda^*$ is called the *Lagrange multiplier vector*, and the vector $Z(x^*)^T g(x^*)$ of (3.1.7) is called the *reduced gradient*. Condition (3.1.6) can also be written as

$$g(x^*) - A(x^*)^T \lambda^* = 0,$$

which can be interpreted as a statement that x^* is a *stationary point* (with respect to x) of the *Lagrangian function*

$$L(x, \lambda) \equiv f(x) - \lambda^T c(x) \tag{3.1.9}$$

when $\lambda = \lambda^*$. Note that λ^* is a stationary point of $L(x, \lambda)$ (with respect to λ) at any feasible point x.

We emphasize that in general the solution x^* of NEP is *not* an unconstrained minimizer of the Lagrangian function. For example, consider the one-variable problem

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \; x^3 \quad \text{subject to} \quad x + 1 = 0, \tag{3.1.10}$$

whose (unique) solution is obviously $x^* = -1$, with Lagrange multiplier $\lambda^* = 3$. However, x^* is an unconstrained *maximizer* of the associated Lagrangian function $L(x, \lambda^*) = x^3 - 3(x + 1)$.

To derive second-order optimality conditions, we examine the behavior of f along feasible arcs from an alleged solution of NEP.

Theorem 3.1.3. *(Second-order necessary optimality conditions.) If the constraint qualification holds at x^*, necessary conditions for x^* to be a minimizer of NEP are:*

1. *$g(x^*) = A(x^*)^T \lambda^*$ for some λ^*; and*
2. *$p^T \nabla^2 L(x^*, \lambda^*) p \geq 0$ for any nonzero vector p satisfying $A(x^*)p = 0$ and for any λ^* satisfying (i).*

Proof. Assume that x^* is a local minimizer of NEP. Part (i) holds by Theorem 3.1.2. Now consider a nonzero vector p satisfying (3.1.8). (If none exists, the theorem holds vacuously.) Let $x(\alpha)$ be the twice-differentiable arc whose existence is guaranteed by the constraint qualification, where $x(0) = x^*$ and $dx(0)/d\alpha = p$. Let v denote

$d^2x(0)/d\alpha^2$, and assume henceforth that all vector and matrix functions are evaluated at x^* unless otherwise specified. Since each constraint function c_i is identically zero along $x(\alpha)$, we have

$$\begin{aligned}\frac{d^2}{d\alpha^2}c_i(x(0)) &= a_i^T\frac{d^2}{d\alpha^2}x(0) + \frac{d}{d\alpha}(a_i^T)\frac{d}{d\alpha}x(0)\\ &= a_i^Tv + p^T\nabla^2c_ip = 0.\end{aligned} \tag{3.1.11}$$

Further, using (3.1.6) and (3.1.8),

$$\begin{aligned}\frac{d}{d\alpha}f(x(0)) &= g^T\frac{d}{d\alpha}x(0) = g^Tp\\ &= \lambda^{*T}Ap = 0.\end{aligned} \tag{3.1.12}$$

Condition (i) implies that x^* is a stationary point of f along the feasible arc. In order for x^* to be a local minimizer, the *curvature* of f along any feasible arc must be nonnegative, i.e., it must hold that

$$\frac{d^2}{d\alpha^2}f(x(0)) \geq 0. \tag{3.1.13}$$

Using (3.1.12), the definition of v, and (3.1.6), we write (3.1.13) as

$$\begin{aligned}\frac{d^2}{d\alpha^2}f(x(0)) &= \frac{d}{d\alpha}(g^T\frac{d}{d\alpha}x(0))\\ &= g^T\frac{d^2}{d\alpha^2}x(0) + p^T\nabla^2fp\\ &= \lambda^{*T}Av + p^T\nabla^2fp \geq 0.\end{aligned} \tag{3.1.14}$$

Rewriting (3.1.11) as $a_i^Tv = -p^T\nabla^2c_ip$ and substituting this expression into (3.1.14), we obtain

$$\begin{aligned}\frac{d^2}{d\alpha^2}f(x(0)) &= -p^T(\sum_{i=1}^m\lambda_i^*\nabla^2c_i)p + p^T\nabla^2fp\\ &= p^T\nabla^2L(x^*,\lambda^*)p \geq 0,\end{aligned}$$

which is the desired result. ∎

A compact statement of condition (ii) of this theorem is that the $(n-m)\times(n-m)$ matrix $Z(x^*)^T\nabla^2L(x^*,\lambda^*)Z(x^*)$ must be positive semi-definite. (This matrix is called the *reduced Hessian of the Lagrangian function.*)

Sufficient conditions for x^* to be a local minimizer of NEP can be similarly derived, and the following theorem will be stated without proof.

Theorem 3.1.4. *(Sufficient conditions for optimality.) A feasible point x^* is a strong local minimizer of NEP if there exists a vector λ^* such that*

1. *$g(x^*) = A(x^*)^T\lambda^*$; and*
2. *$Z(x^*)^T\nabla^2L(x^*,\lambda^*)Z(x^*)$ is positive definite.* ∎

3.2. Overview of Methods

Before considering different classes of methods for solving NEP (3.1.1), we present an overview of the motivation that underlies many seemingly different algorithms. The basic principle invoked in solving NEP is that of *replacing a difficult problem by an easier problem.* Application of this principle leads to methods that formulate and solve a *sequence of subproblems*, where each subproblem is related in a known way to the original problem. In some of the methods to be discussed, the subproblem involves the *unconstrained* minimization of a model function; in other instances, the subproblem includes bounds and/or linear constraints derived from constraints of the original problem.

The most common source of the (simpler) model functions in the subproblems is the Taylor series expansion of a general nonlinear (but smooth) function, which is widely used throughout numerical analysis. For example, Newton's method for zero-finding is based on successively finding the zero of the local linear approximation obtained by including only the first-order term of the Taylor series. When *minimizing*, a linear model is usually inappropriate, since a general linear function is unbounded below. Therefore, minimization is most frequently based on developing a *quadratic* model of the function to be minimized. For example, suppose that the problem is to minimize the unconstrained function $f(x)$, and that x_k is the current iterate. A local quadratic model is obtained by truncating the usual Taylor expansion of f about x_k:

$$f(x_k + p) \approx f(x_k) + g(x_k)^T p + \tfrac{1}{2} p^T H(x_k) p, \tag{3.2.1}$$

where $H(x)$ denotes the (symmetric) Hessian matrix $\nabla^2 f(x)$. If $H(x_k)$ is positive definite, the quadratic model (3.2.1) has a proper minimizer at $x_k + p_k$, where p_k solves the following system of linear equations:

$$H(x_k) p_k = -g(x_k). \tag{3.2.2}$$

Based on this analysis, Newton-based linesearch methods for unconstrained optimization typically define the next iterate as

$$x_{k+1} = x_k + \alpha_k p_k,$$

where α_k is a positive steplength, and the search direction p_k is defined by (3.2.2). The value of α_k is chosen to ensure a decrease in a quantity that measures progress toward the solution. For details, see, e.g., [Ortega and Rheinboldt 1970] or [Dennis and Schnabel 1983].

Because models derived from the Taylor series necessarily neglect higher-order terms, their validity is assured only in a neighborhood (of unknown size) of the current point. Therefore, a standard technique for restricting the region in which the model applies is to add constraints that prevent iterates from moving too far from the current point. For example, the step p_k from x_k to x_{k+1} might be chosen not as the unconstrained minimizer of (3.2.1), but as the minimizer of (3.2.1) subject to a restriction on the size of $\|p\|$. Trust-region methods are based on *explicitly*

restricting the domain in which a model function is considered reliable, and vary depending on the measure and form of the domain restriction.

For nonlinearly constrained problems, the same general principle applies of creating subproblems based on local models of the objective and constraint functions. Approaches to solving NEP vary in the form of the subproblem to be solved at each iteration, and in the definition of any model functions. Further variation is induced by the use of different numerical methods for solving mathematically equivalent problems. The difficulty of solving nonlinear constrained problems has prompted the development of methods that endeavor to transform NEP into that of solving either a single or a sequence of of problems that are *not* nonlinearly constrained. This approach was particularly attractive in the early development of the field when software existed for unconstrained problems, but not for constrained problems. It continued when software for linearly constrained problems became available. Later a similar develop occurred when software for large-scale problems was required. It proved much easier to extend software for unconstrained and linearly constrained problems by transforming NEP than it was to develop software based on direct methods to solve NEP. Moreover, the efficiency of the procedures to solve the linear subproblems that arise was now critical and these subproblems were easier to handle in the methods based on transformations. A further unexpected development was that one class of transformation methods (barrier functions) had the unusual property of avoiding the combinatorial aspect of inequality constrained problems. The net result is that there is a much wider variety of methods to solve nonlinearly constrained problems than to solve simpler problems. We start by discussing one of the earliest transformation methods.

3.3. The Quadratic Penalty Function

3.3.1. Background

Although penalty functions in their original form are not widely used today (perhaps it would be better to have said "should not be widely used today" since they are still highly popular with engineers), a thorough understanding of their properties is important background for more recent methods. For more detailed discussions of penalty-function methods, see, e.g., [Fiacco and McCormick 1968], [Fletcher 1981], [Gill, Murray and Wright 1981] and [Luenberger 1984]. Penalty functions have a long history, occur in many forms, and are often called by special names in different applications. Their motivation is always to ensure that the iterates do not deviate "too far" from the constraints.

In general, an unconstrained minimizer of f will occur at an infeasible point, or f may be unbounded below. Therefore, any algorithm for NEP must consider not only the minimization of f, but also the enforcement of feasibility. In solving NEP, a "natural" strategy is to devise a *composite function* whose *unconstrained* minimizer is either x^* itself, or is related to x^* in a known way. The original problem can then be solved by formulating a sequence of unconstrained subproblems (or possibly a single unconstrained subproblem). Intuitively, this can be achieved by minimizing a function that combines f and a term that "penalizes" constraint violations.

A "classical" penalty term (usually attributed to Courant [1943]) is the squared Euclidean norm of the constraint violations, which leads to the well known *quadratic penalty function*

$$P_Q(x,\rho) \equiv f(x) + \tfrac{1}{2}\rho c(x)^T c(x) = f(x) + \tfrac{1}{2}\rho\|c(x)\|_2^2, \tag{3.3.1}$$

where the nonnegative scalar ρ is called the *penalty parameter*. Let $x^*(\rho)$ denote an unconstrained minimizer of $P_Q(x,\rho)$. The following theorem, which is proved in Fiacco and McCormick [1968], shows that, under reasonably general conditions,

$$\lim_{\rho\to\infty} x^*(\rho) = x^*.$$

Theorem 3.3.1. *(Convergence of the quadratic penalty method.) Let $\{\rho_k\}$ be a strictly increasing unbounded positive sequence. Assume that there exists a nonempty, isolated compact set Ω of local solutions of NEP with the same optimal function value. Then there exists a compact set S such that $\Omega \subset S$, and for sufficiently large ρ_k, there exist unconstrained minimizers of $P_Q(x,\rho_k)$ in the interior of S. Further, every limit point of any subsequence of the minimizing points is in Ω.* ∎

In effect, this result ensures that applying the quadratic penalty-function transformation creates a set of *local minimizers* of the penalty function. For ρ sufficiently large, and within a bounded region including x^*, the sequence of local minimizers of $P_Q(x,\rho)$ will converge to x^*. It may nonetheless happen that the unconstrained algorithm used to minimize $P_Q(x,p)$ will fail to converge to the desired local minimizer. This limitation is illustrated in an example given by Powell [1972]:

$$\underset{x\in\mathbb{R}^1}{\text{minimize}}\ x^3 \quad \text{subject to} \quad x-1=0,$$

whose solution is trivially $x^* = 1$. The associated penalty function is

$$P_Q(x,\rho) = x^3 + \tfrac{1}{2}\rho(x-1)^2, \tag{3.3.2}$$

and has a local minimizer at

$$x^*(\rho) = \frac{-\rho + \sqrt{\rho^2+12\rho}}{6}, \quad \text{with} \quad \lim_{\rho\to\infty} x^*(\rho) = x^* = 1.$$

The function (3.3.2) is unbounded below for any finite ρ, and hence there is no guarantee that an unconstrained algorithm will converge to the minimizer from an arbitrary starting point. While this example illustrates a limitation of the approach most alternative methods also have limitations. The main drawback of the use of peanlty functions is they give rise to a sequence of *difficult* unconstrained problems.

3.3.2. Properties of the quadratic penalty function

The solutions of intermediate unconstrained subproblems have several interesting features. Given a strictly increasing positive unbounded sequence $\{\rho_k\}$, assume that k is large enough so that the result of Theorem 3.3.1 holds, and let $\{x_k^*\}$ denote the local unconstrained minimizer of $P(x, \rho_k)$ nearest to x^*. Let c_k denote $c(x_k^*)$ and f_k denote $f(x_k^*)$. Then the following properties hold:

1. the sequence $\{P(x_k^*, \rho_k)\}$ is nondecreasing;
2. $\{f_k\}$ is nondecreasing;
3. $\{\|c_k\|_2\}$ is nonincreasing.

Thus, in general, each successive x_k^* displays a decreasing measure of infeasibility and an increasing value of the objective function. Further, each x_k^* is a point at which no descent direction exists with respect to both f and $\|c\|_2$.

Since P_Q is a smooth function, its gradient must vanish at the unconstrained minimizer $x^*(\rho)$, so that the following condition holds:

$$g(x^*(\rho)) = -\rho A(x^*(\rho))^T c(x^*(\rho)). \tag{3.3.3}$$

Condition (3.3.3) states that the gradient of the objective function at $x^*(\rho)$ is a linear combination of the constraint gradients, and hence has the same form as the first-order necessary condition for optimality when x^* satisfies the constraint qualification (Theorem 3.1.2), namely:

$$g(x^*) = A(x^*)^T \lambda^*. \tag{3.3.4}$$

Comparing (3.3.3) and (3.3.4), we see that the quantity $\lambda_i(\rho) = -\rho c_i(x^*(\rho))$, $i = 1, \ldots, m$, is analogous to the i-th Lagrange multiplier at x^*. When $A(x^*)$ has full rank and the sufficient conditions of Theorem 3.1.4 hold at x^*, the convergence of $x^*(\rho)$ to x^* and the uniqueness of the Lagrange multipliers imply that

$$\lim_{\rho \to \infty} \lambda_i(\rho) = \lambda_i^*. \tag{3.3.5}$$

Under suitable assumptions, the set of unconstrained minimizers of the penalty function can be regarded as a function of an independent variable, tracing out a smooth *trajectory* of points converging to x^*. The following result is proved in Fiacco and McCormick [1968].

Theorem 3.3.2. *(Smooth trajectory of the quadratic penalty function.) Assume that $A(x^*)$ has full rank, and that the sufficient conditions of Theorem 3.1.4 hold at x^*. Then, for ρ sufficiently large, there exist continuously differentiable functions $x(\rho)$ and $\lambda(\rho)$, such that*

$$\lim_{\rho \to \infty} x(\rho) = x^* \quad \text{and} \quad \lim_{\rho \to \infty} \lambda(\rho) = \lambda^*.$$

Further, $x(\rho)$ is an unconstrained local minimizer of $P(x, \rho)$ for any finite ρ. ∎

The trajectory of minimizers of the quadratic penalty function has several interesting properties. In order to discuss conditions at the limit point, we introduce the variable $r = 1/\rho$, use the notation $x(r) \equiv x(\rho)$, and take limits as r approaches zero. Since Theorem 3.3.2 implies the existence of a smooth function $x(r)$, we expand about $r = 0$:

$$x(r) = x^* + ry + O(r^2),$$

where

$$y \equiv \lim_{r \to 0} \frac{x(r) - x^*}{r} = x'(0) = \left. \frac{dx(r)}{dr} \right|_{r=0}.$$

Because $x(r)$ is an unconstrained minimizer of the quadratic penalty function, for $r > 0$ we have the identity

$$r\nabla P(x(r), r) \equiv rg(r) + A(r)^T c(r) = 0, \tag{3.3.6}$$

where the notation "(r)" denotes evaluation at $x(r)$. Differentiating (3.3.6) with respect to r at $x(r)$ and using the chain rule, we obtain:

$$\frac{d}{dr}(rg + A^T c) = g + rHx'(r) + A^T A x'(r) + \sum_{i=1}^{m} c_i H_i x'(r) = 0, \tag{3.3.7}$$

where H denotes $\nabla^2 f$, H_i denotes $\nabla^2 c_i$, and all functions are evaluated at $x(r)$. As $r \to 0$, we know that $c_i \to 0$, and hence, using (3.3.4), (3.3.7) becomes in the limit:

$$g(x^*) + A(x^*)^T A(x^*) y = A(x^*)^T \lambda^* + A(x^*)^T A(x^*) y = 0. \tag{3.3.8}$$

Equation (3.3.8) implies that $A(x^*)^T(\lambda^* + A(x^*)y) = 0$ and since $A(x^*)$ has full row rank, we may assert that

$$A(x^*)y = -\lambda^*. \tag{3.3.9}$$

If $\lambda_i^* \neq 0$, (3.3.9) implies that the trajectory of minimizers of the quadratic penalty function generates a *nontangential* approach to x^*.

3.3.3. Practical issues

Based on the theory developed thus far, it might appear that the solution to NEP could be found simply by setting ρ to a very large value and using a standard unconstrained method. Unfortunately, the quadratic penalty function has the property that the Hessian matrices $\nabla^2 P(x^*(\rho), \rho)$ become increasingly ill-conditioned as ρ increases (see [Murray 1969], and [Lootsma 1969]). To see why, observe that the Hessian of P_Q at an arbitrary point x is given by:

$$\nabla^2 P_Q(x, \rho) = H + \sum_{i=1}^{m} \rho c_i H_i + \rho A^T A. \tag{3.3.10}$$

At $x^*(\rho)$, for large ρ, it follows from (3.3.5) that the first two terms in (3.3.10) form an approximation to the bounded matrix $\nabla^2 L(x^*, \lambda^*)$. Thus, if $m < n$, the matrix

(3.3.10) evaluated at $x^*(\rho)$ is dominated by the unbounded rank-deficient matrix $\rho A^T A$.

Analysis of the eigenvalues and eigenvectors of $\nabla^2 P(x^*(\rho), \rho)$ as $\rho \to \infty$ (see [Murray 1971b]) reveals that $(n - m)$ eigenvalues are bounded, with associated eigenvectors that in the limit lie in the null space of $A(x^*)$. However, the remaining m eigenvalues are of order ρ, i.e., unbounded in the limit, and the corresponding eigenvectors lie in the range of $A(x^*)^T$. Thus, application of a *general* unconstrained method to minimize $P_Q(x, \rho)$ (i.e., a method that simply solves equations involving $\nabla^2 P$ without taking account of the special structure) is unsatisfactory because the near-singularity of the Hessian matrices will impede local convergence. To overcome this drawback, the search direction can be computed by solving an augmented system similar to (2.1.14) that reflects the very special structure of $\nabla^2 P$ (see [Gould 1986]). This approach is closely related to sequential quadratic programming methods (see Section 3.5). Although the numerical difficulties associated with an ill-conditioned Hessian can be addressed (by this we mean an accurate solution of the equations defining the search direction can be obtained if care is taken) the poor rate of convergence that results when the Hessian is nearly singular at the solution still remains.

3.4. The ℓ_1 Penalty Function

With the quadratic penalty function, the penalty parameter ρ must become infinite in order to achieve convergence to x^*. In contrast, we can devise a *non-differentiable* penalty function of which x^* is the unconstrained minimizer. (Such a function is called an *exact* penalty function.)

The most widely used exact penalty function is the ℓ_1 *penalty function*, or *absolute-value penalty function*:

$$P_1(x, \rho) \equiv f(x) + \rho \sum_{i=1}^{m} |c_i| = f(x) + \rho \|c(x)\|_1, \tag{3.4.1}$$

where $\rho \geq 0$. The ℓ_1 penalty function has been used for many years (under different names) in structural and mechanical design problems.

The function $P_1(x, \rho)$ has discontinuous derivatives at any point where a constraint function vanishes, and hence x^* will be a point of discontinuity in the gradient of P_1. An important difference between P_1 and P_Q (3.3.1) is that, under mild conditions, ρ in (3.4.1) need not become arbitrarily large in order for x^* to be an unconstrained minimizer of P_1. Rather, there is a threshold value $\bar{\rho}$ such that x^* is an unconstrained minimizer of P_1 for *any* $\rho > \bar{\rho}$. Thus, whenever ρ in (3.4.1) is "sufficiently large", an unconstrained minimizer of P_1 will be a solution of the original problem NEP.

For example, consider the problem

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \; x^2 \quad \text{subject to} \quad x - 1 = 0,$$

with solution $x^* = 1$ and multiplier $\lambda^* = 2$. The associated ℓ_1 penalty function is

$$P_1(x, \rho) = x^2 + \rho|x - 1|,$$

of which x^* is an unconstrained minimizer if $\rho > 2$.

Methods based on non-differentiable penalty functions avoid the ill-conditioning associated with P_Q, since the penalty parameter can remain finite. Unfortunately, the crucial value $\bar{\rho}$ depends on quantities evaluated at x^* (which is, of course, unknown), and therefore the value of ρ may need to be adjusted as the algorithm proceeds. Difficulties can arise if an unsuitable value of the penalty parameter is chosen. If ρ is too small, the penalty function may be unbounded below. On the other hand, the unconstrained subproblem will be ill-conditioned if ρ is too large, and special techniques are then required to obtain an accurate solution.

Since P_1 is non-differentiable even at the solution, standard unconstrained methods designed for smooth problems cannot be applied directly. However, special algorithms (see, e.g., [Coleman and Conn 1982a, b]) have been designed for minimizing (3.4.1) that utilize information about the original nonlinearly constrained problem. These methods will be discussed in Section 3.5.5.

Any norm could be used in place of the l_1 norm. For example we could use the l_2 norm. This should not be confused with the quadratic penalty function since the norm requires the square root and like the l_1 norm is nondifferentiable. However, its only nondifferentiable when *all* the elements of $c(x)$ are zero, which is much less common than a single element being zero. Indeed one would expect that a discontinuity is encountered in almost all linesearches when using the l_1 norm while they will almost never occur within a line search when using the l_2 norm.

Using the l_1 norm it can be shown that the required minimizer exists provided ρ is larger than the largest (in magnitude) Lagrange multiplier. However, the size of ρ necessary for a given initial point to be in a compact level set containing the minimizer may be much larger.

3.5. Sequential Quadratic Programming Methods

3.5.1. Motivation

Penalty function methods are based on the idea of combining a weighted measure of the constraint violations with the objective function. In contrast, we now turn to methods based directly on the optimality conditions for problem NEP. The idea of a *quadratic model* is a major ingredient in the most successful methods for unconstrained optimization. However, some care is needed for the nonlinearly constrained case, since we have clearly seen in the derivation of optimality conditions for NEP (Section 3.1) that the important curvature is that of the *Lagrangian function* $L(x, \lambda) = f(x) - \lambda^T c(x)$ (see equation (3.1.9)), and not merely that of f itself. This suggests that our quadratic model should be of the *Lagrangian function*. However, such a model would not be a complete representation of the properties of problem NEP.

Recall from Section 3.1 that x^* is (in general) only a *stationary point* of the Lagrangian function, and *not* an unconstrained minimizer. Even when the sufficient conditions of Theorem 3.1.4 hold, x^* is a minimizer of the Lagrangian function only within the subspace of vectors satisfying $A(x^*)p = 0$. Such a restriction to a subspace suggests that *linear constraints* should be imposed on a quadratic model of the Lagrangian function.

With this in mind, we consider the development of an algorithm of the form

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3.5.1}$$

where p_k is a search direction and α_k is a nonnegative steplength. The search direction p_k is intended to be an estimate of the step from the current iterate x_k to x^*, and thus the optimality conditions at x^* should guide the definition of p_k.

The most obvious property of x^* is that it is *feasible*, i.e., $c(x^*) = 0$. Expanding c in a Taylor series about x_k along a general vector p, we have

$$c(x_k + p) = c_k + A_k p + O(\|p\|^2), \tag{3.5.2}$$

where c_k and A_k denote $c(x_k)$ and $A(x_k)$. Ignoring the quadratic and higher-order terms in (3.5.2), the desired search direction p_k will be the step to a zero of a local *linear* approximation to c if

$$c_k + A_k p_k = 0, \quad \text{or} \quad A_k p_k = -c_k. \tag{3.5.3}$$

The relationship (3.5.3) defines a set of *linear equality constraints* to be satisfied by p_k.

We know from the discussion on linearly constrained problems that the constraints (3.5.3) uniquely determine the portion of p_k in the *range* of A_k^T. Note that (3.5.3) is analogous to the definition of a Newton step to the solution of the (underdetermined) nonlinear equations $c(x) = 0$. If A_k has linearly independent rows, the constraints (3.5.3) are always consistent. However, if the rows of A_k are linearly dependent, (3.5.3) may have no solution.

An important aspect of such methods is that, although to first order the search direction is a step to a zero of the constraints, the right-hand side of (3.5.3) generally becomes zero only in the limit. This property contrasts with the widely used feasible-point methods for *linear* constraints, and arises because of the extreme difficulty of remaining feasible with respect to even a single nonlinear constraint. In fact, the enforced maintenance of feasibility (or even near-feasibility) at every iterate tends almost without exception to produce inefficiency when the constraints display a significant degree of nonlinearity. The effort to remain feasible is thus "wasted" at points that are far from optimal. (In effect, enforcement of feasibility at every iterate means that the algorithm takes very small steps along the curved surface defined by the constraints.)

3.5.2. Formulation of a quadratic programming subproblem

Beyond satisfying the linear constraints (3.5.3), the search direction p_k in (3.5.1) should be defined by minimization of a quadratic model of the Lagrangian function.

By analogy with the EQP subproblem (2.1.6), p_k is taken as the solution of the following equality-constrained quadratic program:

$$\underset{p}{\text{minimize}} \qquad g_k^T p + \tfrac{1}{2} p^T B_k p \tag{3.5.4a}$$

$$\text{subject to} \qquad A_k p = -c_k, \tag{3.5.4b}$$

where g_k is $g(x_k)$, the gradient of f at x_k, and the matrix B_k is intended to represent the *Hessian of the Lagrangian function.* For simplicity, we assume that A_k has full rank.

Let Z_k denote a matrix whose columns form a basis for the null space of A_k, i.e., such that $A_k Z_k = 0$. If $Z_k^T B_k Z_k$ is positive definite, the subproblem (3.5.4) has a unique minimizer p_k. The vector p_k can be expressed conveniently in terms of Z_k and a complementary matrix Y_k, whose columns form a basis for the range space of A_k^T, as

$$p_k = Y_k p_Y + Z_k p_Z, \tag{3.5.5}$$

where $Y_k p_Y$ and $Z_k p_Z$ will be called the *range-space* and *null-space* components of p_k.

The constraints (3.5.4b) completely determine the range-space portion of p_k. Substituting from (3.5.5) into (3.5.4b) gives

$$A_k p_k = A_k(Y_k p_Y + Z_k p_Z) = A_k Y_k p_Y = -c_k,$$

by definition of Y_k and Z_k (since $A_k Y_k$ is nonsingular and $A_k Z_k = 0$).

The null-space portion of p_k is determined by minimization of the quadratic objective function within the appropriate null space, after moving in the range space to satisfy the constraints (3.5.4b). The vector p_Z satisfies the following nonsingular linear system:

$$Z_k^T B_k Z_k p_Z = -Z_k^T g_k - Z_k^T B_k Y_k p_Y. \tag{3.5.6}$$

The Lagrange multiplier μ_k of (3.5.4) satisfies the (compatible) overdetermined system

$$g_k + B_k p_k = A_k^T \mu_k. \tag{3.5.7}$$

The subproblem (3.5.4) has several interesting properties. First, observe that the linear term of the quadratic objective function (3.5.4a) is simply g_k, rather than the gradient of the Lagrangian function. This does not alter the solution p_k of (3.5.4), since multiplication by Z_k^T in (3.5.6) annihilates all vectors in the range of A_k^T. However, taking the linear term as g_k produces the desirable feature that, as $p_k \to 0$, the Lagrange multipliers of the subproblem (3.5.4) will become the Lagrange multipliers of the original nonlinearly constrained problem. Observe that, when x_k is "close" to x^* and $\|p_k\|$ is "small", (3.5.7) becomes arbitrarily close to the first-order optimality condition

$$g(x^*) = A(x^*)^T \lambda^*.$$

The solution p_k and multiplier vector μ_k can be interpreted as the result of a "Newton-like" iteration applied to the set of $n + m$ nonlinear equations that hold at the solution of NEP, namely

$$g(x^*) - A(x^*)^T\lambda^* = 0 \tag{3.5.8}$$

$$c(x^*) = 0. \tag{3.5.9}$$

The constraints $A_k p = -c_k$ define the (underdetermined) Newton step to a point that satisfies (3.5.9). When $B_k = \nabla^2 L(x_k, \lambda_k)$, equation (3.5.7) defines a Newton step in *both* x and λ to a point (x^*, λ^*) that satisfies (3.5.8).

3.5.3. Definition of the Hessian

An obviously important element in formulating the subproblem (3.5.4) is the choice of the matrix B_k, which is intended to approximate the Hessian of the Lagrangian function. (Note that the linear constraints of (3.5.4) do not involve the Lagrangian function.) The "best" choice of B_k is still an open question, particularly in the quasi-Newton case (as we shall mention below), and is the subject of much active research today.

When *exact* second derivatives of f and c are available, an "ideal" choice for B_k near the solution would be $\nabla^2 L(x_k, \lambda_k)$, where x_k and λ_k are the current approximations to x^* and λ^*. With this choice, the "pure" SQP method defined by (3.5.1) with $\alpha_k = 1$ should produce quadratic local convergence in both x and λ. (See, e.g., [Goodman 1985].) However, with $B_k = \nabla^2 L(x_k, \lambda_k)$, the reduced Hessian $Z_k^T B_k Z_k$ may be indefinite, in which case the QP subproblem (3.5.4) has an unbounded solution . Research is actively being carried out on strategies to resolve this situation.

When second derivatives are not available, an obvious approach is to let B_k be a *quasi-Newton* approximation to the Hessian of the Lagrangian function. In this case, the solution of the subproblem (3.5.4) is not a Newton step in both the range and null space. Although the constraints (3.5.4b) (and hence the range-space component of p_k) are independent of B_k, the null-space component of p_k will be based on *approximate* second-derivative information. Because of this disparity in the quality of information, *the constraints tend to converge to zero faster than the reduced gradient.* During the final iterations, the behavior of quasi-Newton SQP methods is typically characterized by the relationship

$$\frac{\|p_Y\|}{\|p_Z\|} \to 0,$$

i.e., the final search directions lie almost wholly in the null space of $A(x^*)$.

In defining B_k as a quasi-Newton approximation, the BFGS formula (see Chapter 1) seems a logical choice for updating an approximation to the Hessian of the Lagrangian function. However, certain complications arise because x^* is *not* necessarily an unconstrained minimizer of the Lagrangian function (see Section 3.1).

Consider a BFGS-like update of the form

$$\bar{B} = B - \frac{Bss^TB}{s^TBs} + \frac{vv^T}{v^Ts}, \tag{3.5.10}$$

where a barred quantity is "new", $s = \bar{x} - x$, and v is a vector to be chosen. Since B is intended to approximate the Hessian of the Lagrangian function, a "natural" choice for v in (3.5.10) would be y_L, the change in gradient of the Lagrangian function, i.e.

$$y_L = \bar{g} - g - (\bar{A}^T - A^T)\lambda, \tag{3.5.11}$$

with λ the best available multiplier estimate. However, it may be *impossible*, with *any* linesearch, to find a steplength α_k in (3.5.1) such that y_L^Ts is positive (see Section 3.5.4, below). Since the updated matrix $\bar{B}$ will be positive definite only if $v^Ts > 0$, performing the update with $v = y_L$ as in (3.5.11) would lead to an indefinite Hessian approximation.

The question thus arises of what to do under these circumstances. If the update is *skipped* when $y_L^Ts < 0$, *no new information* about curvature of the Lagrangian function will be gained from this iteration (or possibly from *any* iteration), and favorable local convergence properties of the quasi-Newton method are unlikely to apply. Therefore, a popular method for dealing with this difficulty is to use $v = y_L$ to perform the update (3.5.10) when y_L^Ts is sufficiently positive; otherwise, v is taken as a perturbed vector $\bar{y}_L$ such that $\bar{y}_L^Ts > 0$. (Such a strategy was first suggested by Powell [1978].) For constrained problems, a necessary condition for superlinear convergence [Boggs, Tolle and Wang 1982] is that the approximate Hessian matrices must satisfy

$$\lim_{k\to\infty} \frac{\|Z_kZ_k^T(B_k - \nabla^2L(x^*,\lambda^*))Z_kZ_k^Tp_k\|}{\|p_k\|} = 0. \tag{3.5.12}$$

The definition of $\bar{y}_L$ should ensure that (3.5.12) is satisfied as the solution is approached, so that superlinear convergence is not inhibited by the update.

3.5.4. Choice of the steplength; merit functions

A steplength α_k is included in the definition (3.5.1) of the SQP iteration in order to ensure "progress" at every iteration, since the current approximation of the Lagrangian function and/or the constraints may be inaccurate when the current iterate is far from x^*. In linesearch methods for unconstrained and linearly constrained optimization, the value of the objective function f alone provides a "natural" measure to guide the choice of α_k. Not too surprisingly, matters are much more complicated when solving a nonlinearly constrained problem. Except in a few special cases, it is impossible to generate a feasible sequence of iterates with decreasing values of the objective function.

The most common approach is to choose α_k in (3.5.1) to yield a "sufficient decrease" (in the sense of Ortega and Rheinboldt [1970]) in a *merit function* M that measures progress toward the solution of NEP. Typically, a merit function is a combination of the objective and constraint functions. An "ideal" merit function should have certain properties, some more important than others. An *essential* property

is that it should always be possible to achieve a sufficient decrease in M when the search direction is defined by the QP subproblem (3.5.4). A desirable feature is that the merit function should not restrict the "natural" rate of convergence of the SQP method, e.g., if $B_k = \nabla^2 L(x_k, \lambda_k)$, then $\alpha_k = 1$ should be accepted at all iterations "near" the solution, in order to achieve quadratic convergence (see Section 3.5.2). An intuitively appealing feature is that x^* should be an unconstrained minimizer of M. A feature with great practical importance is that calculation of M should not be "too expensive" in terms of evaluations of the objective and constraint functions and/or their gradients.

A commonly used merit function is the ℓ_1 *penalty function* (see Section 3.4):

$$M_1(x, \rho) = f(x) + \rho \|c(x)\|_1, \tag{3.5.13}$$

where ρ is a nonnegative penalty parameter. (Han [1977] first suggested use of this function as a means of "globalizing" an SQP method.) This merit function has the property that, for ρ sufficiently large, x^* is an unconstrained minimizer of $M_1(x, \rho)$. In addition, ρ can always be chosen so that the SQP search direction p_k is a descent direction for $M_1(x, \rho)$. However, requiring a decrease in M_1 at every iteration can lead to the inhibition of superlinear convergence (the "Maratos effect"; see [Maratos 1978]), and various strategies have been devised to overcome this drawback (see, e.g., [Chamberlain *et al.* 1982]). In practice, the choice of penalty parameter in (3.5.13) can have a substantial effect on efficiency.

An increasingly popular alternative is the *augmented Lagrangian function*:

$$M_A(x, \lambda, \rho) \equiv f(x) - \lambda^T c(x) + \tfrac{1}{2}\rho c(x)^T c(x), \tag{3.5.14}$$

where λ is a multiplier estimate and ρ is a nonnegative penalty parameter (see Sections 3.3 and 3.7). Use of (3.5.14) as a merit function was suggested by Wright [1976] and Schittkowski [1981]. If λ in (3.5.14) is the optimal multiplier vector λ^*, then x^* is a stationary point (with respect to x) of M_A (see Section 3.1). As with M_1 (3.5.13), it can be shown that there exists a *finite* $\bar{\rho}$ such that x^* is an *unconstrained minimizer* of $M_A(x, \lambda^*, \rho)$ for all $\rho > \bar{\rho}$. With suitable choice of λ, (3.5.14) does not impede superlinear convergence.

Many subtle points need to be studied in using (3.5.14) as a merit function—in particular, extreme care must be exercised in defining the multiplier estimate λ. If λ is taken simply as the "latest" multiplier estimate (i.e., the multiplier vector of the most recent QP subproblem (3.5.4)), the merit function changes *discontinuously* at every iteration, and difficulties consequently arise in proving global convergence. To avoid this situation, the vector λ can be treated as an *additional unknown*, which is then included in the linesearch. Typically, the QP multipliers μ_k are used to define a multiplier "search direction" ξ_k, so that $\xi_k = \mu_k - \lambda$. (See, e.g., [Tapia 1977], [Schittkowski 1981], [Gill, Murray, Saunders and Wright 1986].)

The most successful implementations of SQP methods for problems in which only first derivatives are available typically use a modified BFGS update to define B_k (see (3.5.10)), and either M_1 (3.5.13) or M_A (3.5.14) as a merit function.

3.5.5. Related methods

It is often difficult to classify methods because the chosen hierarchical structure tends to be subjective. (Other authors may consider that SQP methods are subsumed under a category defined by the methods of this section!) We take the general view that an "SQP method" includes a subproblem derived from the optimality conditions of Theorem 3.1.3, with linearized versions of the nonlinear constraints, and a quadratic objective function whose Hessian reflects the curvature of the Lagrangian function. The methods to be discussed are considered as "related" to SQP methods because they are derived in one sense or another from the optimality conditions. However, the subproblem may not have the precise form stated above. In some cases, the subproblem may be *equivalent* to a quadratic program, but with modified objective function or constraints.

Two obvious deficiencies of any SQP method defined by (3.5.4) are that the constraints (3.5.4b) may be inconsistent if A_k does not have full row rank, and that the search direction may "blow up" in size if the rows of A_k are "nearly" linearly dependent. Several modifications to the basic SQP structure are designed to correct these difficulties.

Fletcher [1981, 1985] has suggested a class of methods called "Sℓ_1QP methods" in which the search direction p_k is the solution of the following subproblem:

$$\begin{array}{ll} \underset{p}{\text{minimize}} & g_k^T p + \frac{1}{2} p^T H p + \rho \|A_k p + c_k\|_1 \\ \text{subject to} & \|p\|_\infty \leq \beta, \end{array} \tag{3.5.15}$$

where ρ is a penalty parameter and β is a positive number. The benefit of this formulation is that a solution to (3.5.15) always exists, even when the linearized constraints (3.5.4b) are inconsistent. Further, the ℓ_1 penalty term in the objective and the explicit bound ("trust-region") constraints on each component of p ensure that the search direction and multiplier vector are bounded.

We characterize this approach as "SQP-related" because subproblem (3.5.15) is equivalent to the following quadratic program with inequality constraints:

$$\begin{array}{ll} \underset{p}{\text{minimize}} & g_k^T p + \frac{1}{2} p^T H p + \rho e^T (u + v) \\ \text{subject to} & -\beta e \leq p \leq \beta e, \\ & A_k p + c_k = u - v, \quad u \geq 0, \quad v \geq 0, \end{array} \tag{3.5.16}$$

where e denotes the vector $(1, 1, \ldots, 1)^T$ of appropriate dimension. Methods for solving quadratic programs with inequality constraints such as (3.5.16) will be discussed in Section 3.9.

Other approaches have been suggested that are closely related to SQP methods, although derived from a different perspective. For example, Coleman and Conn [1982a, b] suggest a method based on unconstrained minimization of the ℓ_1 penalty function of Section 3.5. The similarity to an SQP method arises because the search is computed as two orthogonal components that lie in the range space of A_k^T and null space of A_k, with the range-space component based on linearization of the nonlinear constraints.

SQP and SQP-related methods are widely considered the most effective general methods today for solving NEP. Several have been implemented in highly reliable software, and perform extremely well in practice, even on test problems formerly regarded as difficult.

3.6. Sequential Linearly Constrained Methods

The methods to be discussed in this section—*reduced Lagrangian* or *sequential linearly constrained (SLC)* methods—were originally devised by Robinson [1972] and Rosen and Kreuser [1972]. They have tended to be most widely used for *large-scale* optimization problems (e.g., in the well known MINOS code; see [Murtagh and Saunders 1982, 1983]). The motivation for an SLC method is the same as that of an SQP method: to minimize the *Lagrangian function* subject to linearizations of the original nonlinear constraints. In contrast to an SQP method, which develops a quadratic model of the Lagrangian function, an SLC method solves a linearly constrained subproblem in which the objective function is a *general* approximation to the Lagrangian function.

A typical subproblem in an SLC method applied to a problem with equality constraints has the form:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \mathcal{F}_k(x) \\ \text{subject to} & A_k(x - x_k) = -c_k, \end{array} \tag{3.6.1}$$

where $\mathcal{F}_k(x)$ is a general approximation to the Lagrangian function at x_k, based on a current multiplier estimate λ_k. Let x_{k+1} be the solution of the subproblem (3.6.1).

Ideally, an SLC method would choose $\mathcal{F}_k(x)$ in (3.6.1) as

$$\mathcal{F}_k(x) = f(x) - \lambda_k^T \bar{c}_k(x), \tag{3.6.2}$$

where λ_k is the "latest" multiplier estimate, and

$$\bar{c}_k(x) = c(x) - c_k - A_k(x - x_k). \tag{3.6.3}$$

(The function $\bar{c}_k(x)$ in (3.6.3) is the difference between $c(x)$ and its linearization at x_k.) With this choice of $\mathcal{F}_k$, the multiplier vector of (3.6.1) will converge to λ^* if x_k converges to x^*. For this reason, λ_k in (3.6.2) is typically taken as the multiplier vector from the previous subproblem (3.6.1).

The SLC method defined by (3.6.1) and (3.6.2) has extremely favorable local convergence properties. (For details, see [Robinson 1972].) If x_0 and λ_0 are "sufficiently close" to x^* and λ^*, and each subproblem (3.6.1) is solved exactly, the sequence of solutions (x_k, λ_k) converges *quadratically* to (x^*, λ^*). Unfortunately, this convergence result is *purely local*, and the iterates may diverge outside a small neighborhood of x^*. For this reason, various strategies have been proposed to improve the reliability of SLC methods, such as executing a few iterations of a penalty function method (Section 3.3) to determine a "good" initial point [Rosen 1978]. Alternatively, Murtagh and Saunders [1982] give an algorithm in which $\mathcal{F}_k(x)$ in (3.6.1) is taken as the following close relative of an *augmented* Lagrangian function:

$$\mathcal{F}_k(x) = f(x) - \lambda_k^T \bar{c}_k(x) + \tfrac{1}{2}\rho \bar{c}_k(x)^T \bar{c}_k(x),$$

where $\bar{c}_k$ is defined by (3.6.3), and the *penalty parameter* ρ is a nonnegative scalar (*cf.* (3.5.14)). The penalty parameter can be adjusted to attain the ideal quadratic convergence rate.

Subproblem (3.6.1) involves the minimization of a general nonlinear function subject to linear equality constraints. The choice of solution method for a given problem will depend on the information available about the problem functions (i.e., the level and cost of derivative information), and on the problem size. Because several iterations may be required to solve the subproblem, SLC methods have a "two-level" structure of *major* and *minor* iterations. Each major iteration involves formulation of a new subproblem, using the current values of x_k and λ_k; the minor iterations are then those of the method used to solve the particular subproblem.

In general, an SLC method tends to require more evaluations of the problem functions than an SQP method to solve the same nonlinearly constrained problem. However, the solution is typically found by an SLC method in fewer *major* iterations, where a major iteration in an SQP method simply involves solving the QP subproblem (3.5.4). The reason that SLC methods are so widely used for large-scale problems is that, somewhat surprisingly, general-purpose techniques for solving large-scale linearly constrained problems are better developed today than general methods for large-scale quadratic programming. However, this situation is likely to change during the next few years. Even in the most advanced SLC methods today, unresolved issues remain concerning proofs of global convergence, the use of a merit function, the definition of $\mathcal{F}_k$, and strategies for early termination of unpromising subproblems.

A source of inefficiency in an SLC method is that the effort required to solve (3.6.1) accurately may be excessive with respect to the improvement gained in the current iterate. If x_k is far from optimal and/or λ_k is a poor estimate of λ^*, overall efficiency may be improved by terminating the minor iterations before (3.6.1) has been solved. Such strategies remain an open research question, since they include a delicate balance between possible gains in speed and loss of theoretical convergence properties.

3.7. Augmented Lagrangian Methods

The class of *augmented Lagrangian methods* can be derived from several different viewpoints. In all cases, a fundamental aim is to use the optimality conditions to devise a well-behaved *unconstrained* subproblem of which x^* is the solution. Augmented Lagrangian methods became extremely popular in the early 1970's, largely because of the great success of methods for unconstrained optimization. This approach to nonlinear programming was suggested independently by Hestenes [1969] and Powell [1969].

Assume that the sufficient conditions for optimality (Theorem 3.1.4) hold at x^*. Then x^* is a *stationary point* of the Lagrangian function $L(x, \lambda) = f(x) - \lambda^T c(x)$, when $\lambda = \lambda^*$ (see Section 3.1). Because x^* is not necessarily a *minimizer* of the Lagrangian function, the Lagrangian function itself is not a suitable choice for the objective function of the subproblem, even if λ^* were known.

Since the *reduced* Hessian of the Lagrangian function is positive definite, x^* is a minimizer of the Lagrangian function within the subspace of vectors orthogonal to $A(x^*)$. (This observation was the primary motivation for SQP and SLC methods; see Sections 3.5 and 3.6.) Positive-definiteness of the *reduced* Hessian of the Lagrangian function indicates that the Lagrangian function can display negative curvature at x^* *only along directions in the range space of* $A(x^*)^T$. This suggests that a suitable function for an unconstrained subproblem might be obtained by *augmenting* the Lagrangian function through addition of a term that retains the stationary properties of x^*, but alters the Hessian in the range space of $A(x^*)^T$.

The most popular such *augmented Lagrangian function* is obtained by adding a quadratic penalty term, which gives

$$L_A(x, \lambda, \rho) \equiv f(x) - \lambda^T c(x) + \tfrac{1}{2}\rho c(x)^T c(x), \tag{3.7.1}$$

where ρ is a nonnegative penalty parameter. Both the quadratic penalty term of (3.7.1) and its gradient vanish at x^*. Thus, if $\lambda = \lambda^*$, x^* is a stationary point (with respect to x) of (3.7.1). The Hessian matrix of the augmented Lagrangian function is

$$\nabla^2 L_A(x, \lambda, \rho) = \nabla^2 f(x) - \sum_{i=1}^{m} (\lambda_i - \rho c_i(x)) \nabla^2 c_i(x) + \rho A(x)^T A(x).$$

Since $c(x^*) = 0$, the Hessian of the penalty term at x^* is simply $\rho A(x^*)^T A(x^*)$, which is a positive semi-definite matrix with strictly positive eigenvalues corresponding to eigenvectors in the range of $A(x^*)^T$. Thus, the presence of the penalty term in L_A has the effect of increasing the (possibly negative) eigenvalues of $\nabla^2 L(x^*, \lambda^*)$ corresponding to eigenvectors in the range space of $A(x^*)^T$, but leaving the other eigenvalues unchanged. Using this property, under mild conditions there exists a *finite* $\bar{\rho}$ such that x^* is an *unconstrained minimizer* of $L_A(x, \lambda^*, \rho)$ for all $\rho > \bar{\rho}$. In example (3.1.10), the crucial value is $\bar{\rho} = 6$, since $x^* = -1$ is a (local) unconstrained minimizer of $L_A(x, \lambda^*, \rho) = x^3 - 3(x+1) + \frac{1}{2}\rho(x+1)^2$ if $\rho > 6$.

In a typical augmented Lagrangian method, x_k is taken as the unconstrained minimizer of L_A in (3.7.1), where λ is taken as λ_k, the latest multiplier estimate. Strategies must therefore be developed for choosing both λ_k and ρ.

Although the penalty parameter in (3.7.1) need not become infinite, this restriction is of little practical value in actually choosing a specific value of ρ. Obviously, ρ must be large enough so that x^* is a local minimizer of L_A. However, there are difficulties with either a too-large or a too-small value of ρ. The phenomenon of an ill-conditioned subproblem occurs if ρ becomes too large, as with the quadratic penalty function (Section 3.3). However, if the current ρ is too small, the augmented function may be unbounded, or the Hessian matrix of L_A may be ill-conditioned because ρ is too close to the critical value $\bar{\rho}$ at which the Hessian of L_A is singular.

Since x^* is not a stationary point of L_A except when $\lambda = \lambda^*$, an augmented Lagrangian method will converge to x^* only if the associated multiplier estimates converge to λ^*. Furthermore, when x_k is defined by minimizing (3.7.1), the rate of convergence of x_k to x^* is restricted to the rate of convergence of λ_k to λ^* (see, e.g., [Fletcher 1974]). This result is quite significant, since it implies that

even a quadratically convergent technique applied to determine the unconstrained minimizer of (3.7.1) will not converge quadratically to x^* unless λ_k also converges quadratically to λ^*. (In contrast, the rate of convergence of x_k is not restricted in this fashion in SQP and SLC methods; see Sections 3.5 and 3.6.)

As with an SLC method, it could be argued that the effort required to solve a general unconstrained subproblem to full accuracy may be unjustified if the penalty parameter is wildly out of range, or the Lagrange multiplier estimate is significantly in error. Hence, various strategies have been developed for prematurely terminating the solution of an unpromising subproblem.

3.8. Inequality Constraints

We now derive optimality conditions for a problem in which all constraints are assumed to be *inequalities*:

$$\text{NIP} \qquad \begin{array}{ll} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) \geq 0, \end{array} \tag{3.8.1}$$

where $c(x)$ has m_N components $c_i(x)$, and f and $\{c_i(x)\}$ are twice-continuously differentiable. The matrix $\mathcal{A}(x)$ will denote the Jacobian matrix of the constraint vector $c(x)$, and a solution of NIP will be denoted by x^*.

The derivation of optimality conditions for inequality constraints is more complicated than for equalities for two reasons: first, in general only a subset of the constraints are involved in some of the optimality conditions; and second, the set of feasible perturbations is much larger. The following definitions indicate the increased complexity of terminology.

Definition 3.8.1. *The point $\widehat{x}$ is said to be* feasible *with respect to the inequality constraint $c_i(x) \geq 0$ if $c_i(\widehat{x}) \geq 0$. (Equivalently, the constraint is* satisfied *at $\widehat{x}$.) The constraint $c_i(x) \geq 0$ is said to be* active *at $\widehat{x}$ if $c_i(\widehat{x}) = 0$ and* inactive *if $c_i(\widehat{x}) > 0$. If $c_i(\widehat{x}) < 0$, $\widehat{x}$ is* infeasible*, and the constraint is said to be* violated *at $\widehat{x}$.*

The definition of a local solution of problem NIP is identical to that for problem NEP (Definition 3.1.2), using the appropriate definition of feasibility. For problem NIP, the active constraints at an alleged solution have special importance because they restrict feasible perturbations. If a constraint is inactive at the point x, then it will remain inactive for *any* perturbation in a sufficiently small neighborhood. However, an *active* constraint may be violated by certain perturbations. As in the equality-constraint case, we now consider conditions under which it is possible to characterize feasible perturbations, and accordingly define two constraint qualifications (see [Fiacco and McCormick 1968]). Let $\widehat{c}(x)$ denote the subset of constraints that are *active* at x, and $A(x)$ the Jacobian of $\widehat{c}$. We emphasize that $A(x)$ includes only the gradients of the *active* constraints, whereas $\mathcal{A}(x)$ is the full Jacobian of all the constraints. We shall use the notation $a_i(x)$ for the gradient of an active or inactive constraint, where the meaning should always be clear from the context.

Definition 3.8.2. *(First-order constraint qualification for inequality constraints.) The first-order constraint qualification with respect to the set of inequality constraints $c(x) \geq 0$ holds at the feasible point x if, for any nonzero vector p such that $A(x)p \geq 0$, p is tangent to a differentiable feasible arc emanating from x and contained in the feasible region.*

Definition 3.8.3. *(Second-order constraint qualification for inequality constraints.) The second-order constraint qualification with respect to the inequalities $c(x) \geq 0$ holds at the feasible point x if, for any nonzero vector p such that $A(x)p = 0$, p is tangent to a twice-differentiable arc along which $\widehat{c}$ is identically zero.*

In contrast to the single constraint qualification for equality constraints, these conditions are distinct, and neither implies the other. A condition that ensures satisfaction of both constraint qualifications is given in the following theorem. (See [Fiacco and McCormick 1968] for a proof.)

Theorem 3.8.1. *(Regular point for inequality constraints.) The first- and second-order constraint qualifications for inequalities hold at the feasible point x if $A(x)$ has full rank, i.e., if the gradients of the active constraints are linearly independent.* ∎

We now consider deriving optimality conditions for problem NIP. First, observe from the Taylor expansion (3.1.3) that if $c_i(x) = 0$, i.e., if the i-th constraint is active at x, the constraint becomes *inactive* along a perturbation p such that $a_i^T p > 0$. The optimality theorems utilize the following well known result. (For further discussion and details, see, e.g., [Fletcher 1981].)

Lemma 3.8.1. *(Farkas' Lemma.) Given an $m \times n$ matrix C, a given n-vector b can be expressed as a nonnegative linear combination of the rows of C if and only if, for every vector y such that $Cy \geq 0$, it also holds that $b^T y \geq 0$, i.e.,*

$$b = C^T \lambda, \quad \lambda \geq 0 \quad \textit{if and only if} \quad Cy \geq 0 \Rightarrow b^T y \geq 0. \quad \blacksquare$$

An alternative statement of this result is that a vector z exists such that $b^T z < 0$ and $Cz \geq 0$ if and only if there exists no nonnegative vector λ such that $b = C^T \lambda$. Farkas' Lemma can then be used to prove the following theorem.

Theorem 3.8.2. *(First-order necessary optimality conditions.) If the first-order constraint qualification for inequalities holds at x^*, a necessary condition for x^* to be a minimizer of NIP is that there exists a vector λ^* such that*

$$g(x^*) = A(x^*)^T \lambda^*, \quad \textit{with} \quad \lambda^* \geq 0. \tag{3.8.2}$$

Proof. Assume that x^* is a solution of NIP, and consider any nonzero vector p satisfying

$$A(x^*)p \geq 0. \tag{3.8.3}$$

The first-order constraint qualification implies that p is tangent to a feasible arc emanating from x^* and contained in the feasible region. If $g(x^*)^T p \geq 0$, then Farkas' Lemma immediately implies the existence of $\lambda^* \geq 0$ satisfying (3.8.2). Therefore, we now suppose that there exists a nonzero p satisfying (3.8.3), but such that

$$g(x^*)^T p < 0. \tag{3.8.4}$$

The rate of change of f along the associated feasible arc is $g(x^*)^T p$, so that (3.8.4) implies that f is strictly less than $f(x^*)$ at feasible points in every neighborhood of x^*, thereby contradicting the assumption that x^* is a local minimizer. Thus, there can be no vector p satisfying (3.8.3) and (3.8.4), and Farkas' Lemma implies that (3.8.2) holds. ∎

The crucial difference from the analogous theorem for equality constraints is that *the Lagrange multipliers corresponding to active inequality constraints must be nonnegative.*

In some circumstances, it is useful to define a Lagrange multiplier for *every* constraint, with the convention that the multiplier corresponding to an inactive constraint is zero. Let ℓ^* denote the "extended" multiplier vector. The necessary condition (3.8.2) then becomes

$$g(x^*) = \mathcal{A}^T(x^*)\ell^*, \qquad \ell^* \geq 0, \tag{3.8.5a}$$

where

$$\ell_i^* c_i(x^*) = 0, \quad i = 1, 2, \ldots, m. \tag{3.8.5b}$$

(Condition (3.8.5b) is often called a *complementarity* condition.)

As in the equality case, condition (3.8.5a) implies that x^* is a *stationary point* (with respect to x) of the Lagrangian function, which can be expressed either in terms of the active constraints or all the constraints. We shall use the same notation for both Lagrangian functions—i.e.,

$$L(x, \lambda) \equiv f(x) - \lambda^T \widehat{c}(x), \qquad \text{and} \qquad L(x, \ell) \equiv f(x) - \ell^T c(x).$$

The second-order necessary condition for inequality constraints analogous to Theorem 3.1.3 involves both constraint qualifications.

Theorem 3.8.3. *(Second-order necessary conditions for optimality.) If the first- and second-order constraint qualifications hold at x^*, a necessary condition for x^* to be a local minimizer of NIP is that, for every nonzero vector p satisfying*

$$A(x^*)p = 0, \tag{3.8.6}$$

it holds that $p^T \nabla^2 L(x^*, \lambda^*)p \geq 0$ for all λ^* satisfying (3.8.2).

Proof. Assume that x^* is a minimizer of NIP. Since the first-order constraint qualification holds at x^*, the existence of Lagrange multipliers satisfying (3.8.2) is implied by Theorem 3.8.2. Now consider any nonzero vector p satisfying (3.8.6). Because of the second-order constraint qualification, p is tangent to a feasible arc along which the constraints active at x^* remain identically zero. Exactly as in the proof of Theorem 3.1.3, analysis of the curvature of f along the arc implies the desired result. ∎

Sufficient conditions for x^* to be a local minimizer of NIP are given in the following theorem (see, e.g., [Fiacco and McCormick 1968]):

Theorem 3.8.4. *(Sufficient optimality conditions.) The feasible point x^* is a strong local minimizer of NIP if there exists a vector λ^* such that*

1. $g(x^*) = A^T(x^*)\lambda^*$;
2. $\lambda^* > 0$;
3. $Z(x^*)^T \nabla^2 L(x^*, \lambda^*) Z(x^*)$ *is positive definite, where $Z(x^*)$ is a basis for the null space of $A(x^*)$.* ∎

Condition (ii) of Theorem 3.8.4—that the Lagrange multipliers corresponding to active constraints are strictly positive—is usually termed *strict complementarity*. If any multiplier corresponding to an active constraint is zero, the optimality conditions become more complicated. (For details, see, e.g., [Fiacco and McCormick 1968].)

Before considering methods for the general problem NIP, we turn to the special case of quadratic programming.

3.9. Inequality-Constrained Quadratic Programmming

The inequality-constrained quadratic programming problem is that of minimizing a quadratic function subject to a set of linear inequality constraints:

$$\text{IQP} \qquad \begin{array}{ll} \underset{x}{\text{minimize}} & d^T x + \frac{1}{2} x^T H x \\ \text{subject to} & \mathcal{A}x \geq b. \end{array} \tag{3.9.1}$$

For overviews of quadratic programming, see, e.g., [Fletcher 1981], [Gill, Murray and Wright 1981] and [Fletcher 1986]. Let $g(x)$ denote $d + Hx$, the gradient of the quadratic objective function.

Using the results of Section 3.8, the following conditions must hold if x^* is a minimizer of (3.9.1), where μ^* includes a multiplier for every constraint:

$$\begin{aligned} \mathcal{A}x^* &\geq b \\ g(x^*) = d + Hx^* &= \mathcal{A}^T\mu^* \\ \mu^* &\geq 0 \\ \mu_i^*(a_i^T x^* - b_i) &= 0. \end{aligned} \tag{3.9.2}$$

Let A denote the matrix of constraints active at x^*, and let Z denote a matrix whose columns span the null space of A, i.e., such that $AZ = 0$; then Z^THZ must be positive semi-definite.

The most popular approach to solving IQP is to use a so-called *active-set strategy*, which is based on the following idea. If a feasible point and the correct active set A were known, the solution could be computed directly as described above in the discussion of EQP. Since these are unknown, we develop a *prediction* of the active set—called the *working set*—that is used to compute the search direction, and then change the working set as the iterations proceed.

We shall illustrate the steps of a QP method for a *primal-feasible active-set method*. An initial feasible (non-optimal) point x is required such that $\mathcal{A}x \geq b$. Let A_W (the *working set*) denote a linearly independent set of constraints that are satisfied exactly at the current iterate, and b_W the corresponding components of b, so that $A_W x = b_W$. Let Z_W denote a basis for the null space of A_W, and assume that $Z_W^T H Z_W$ is positive definite. (The treatment of the indefinite and semidefinite cases is complicated, and will not be discussed here.)

A *search direction* p is computed by solving (2.1.12) with $A = A_W$, after which two situations are possible. The point $x + p$ may *violate* a constraint (or several constraints) not currently in the working set. (In this case, A_W is not the correct active set.) In order to remain feasible, a nonnegative step $\bar{\alpha} < 1$ is determined such that $\bar{\alpha}$ is the largest step that retains feasibility. A constraint that becomes satisfied exactly at $x + \bar{\alpha}p$ is then "added" to the working set (i.e., A_W includes a new row), and a new search direction is computed with the modified working set.

Otherwise, the feasible point $x + p$ is the minimizer of the quadratic objective function with the working set treated as a set of equality constraints. Let $\bar{x} = x + p$, and note that $Z_W^T g(\bar{x}) = 0$, which implies that $g(\bar{x}) = A_W^T \mu_W$ for some vector μ_W. If μ_W is nonnegative, then $\bar{x}$ is the solution of IQP, since conditions (3.9.2) are satisfied and $Z_W^T H Z_W$ is positive definite by assumption. Otherwise, there is at least one strictly negative component of μ_W (say, the i-th), and hence there exists a feasible descent direction p, such that $g(\bar{x})^T p < 0$ and $A_W p = e_i$, where e_i is the i-th column of the identity matrix. Movement along p causes the i-th constraint in the working set to become strictly satisfied, and hence effectively "deletes" the constraint from the working set.

Methods of this general structure will converge to a local solution of IQP in a finite number of iterations if at every iteration the active set has full rank, $Z_W^T H Z_W$ is positive definite, and $\mu_W \neq 0$. For details concerning methods that treat more complex situations, see the references given at the beginning of this section.

3.10. Penalty-Function Methods for Inequalities

In applying a penalty-function approach to problem NIP, the motivation is identical to that in the equality-constrained case, namely to add a weighted penalty for infeasibility. Thus, the quadratic or absolute value penalty function may be used as in Sections 3.2 and 3.3, but only the *violated* constraints are included in the penalty term. The quadratic and absolute value penalty functions for NIP may be written

as

$$P_Q(x,\rho) = f(x) + \tfrac{1}{2}\rho\|\widehat{c}(x)\|_2^2 \qquad \text{and} \qquad P_1(x,\rho) = f(x) + \rho\|\widehat{c}(x)\|_1,$$

where $\widehat{c}(x)$ is the vector of constraints *violated* at x.

The convergence of the quadratic and absolute value penalty function methods applied to NIP can be proved as for the equality-constrained problem. (See Sections 3.3 and 3.4.) In an implementation of a penalty-function method, a small tolerance is usually included in the definition of "violated" to avoid discontinuities in the second derivatives of P_Q at x^*, so that, for example:

$$\widehat{c}(x) = \{c_i(x) \mid c_i(x) \le \epsilon\}, \tag{3.10.1}$$

where ϵ ($\epsilon > 0$) is "small".

For P_Q, the existence of a trajectory $x^*(\rho)$, with a multiplier function $\lambda(\rho)$, can be demonstrated as in the equality case, subject to the assumption of strict complementarity (see Sections 3.3 and 3.8). An important property of penalty-function methods can be derived from the behavior of the Lagrange multiplier estimates obtainable at $x^*(\rho)$. (See (3.3.5).) Since

$$-\rho c_i(x^*(\rho)) \to \lambda_i^* \quad \text{and} \quad \lambda_i^* \ge 0,$$

it is clear that a constraint active at x^* with a positive multiplier will be strictly violated at $x^*(\rho)$ for sufficiently large ρ. This relationship explains why the violated set of constraints (3.10.1) is often taken as a prediction of the active set when using a quadratic penalty function method.

3.11. Sequential Quadratic Programming Methods

The *general* motivation given in Section 3.5 for sequential quadratic programming (SQP) methods is unaffected by the change from equality to inequality constraints. Because the optimal point x^* for NIP is a minimizer of the Lagrangian function within the subspace defined by the active constraint gradients (Theorem 3.8.3), the ideas of developing a quadratic model of the Lagrangian function and of linearizing the nonlinear constraints carry over directly. However, this approach will succeed only if we are somehow able to *identify the correct active set*, which is essential in defining the Lagrangian function.

An obvious strategy is to formulate a quadratic programming subproblem like (3.5.4), but with the linear constraints generalized to *inequalities*, to reflect the nature of the nonlinear constraints. Accordingly, the most common SQP method for NIP retains the standard form (3.5.1) for each iteration. The search direction p_k is taken as the solution of the following *inequality-constrained* quadratic program:

$$\underset{p}{\text{minimize}} \qquad g_k^T p + \tfrac{1}{2} p^T B_k p \tag{3.11.1a}$$

$$\text{subject to} \qquad \mathcal{A}_k p \ge -c_k, \tag{3.11.1b}$$

where g_k is $g(x_k)$, the gradient of f at x_k, and the matrix B_k is intended to represent the Hessian of the Lagrangian function. The crucial difference from the equality-constrained case is that the linear constraints of (3.11.1) are *inequalities* involving all the constraints of the original problem.

At the solution of the QP (3.11.1), a subset of the constraints (3.11.1b) will be active. Let A_k denote the subset of active constraints at the solution of (3.11.1), and $\widehat{c}_k$ the corresponding constraint values, so that

$$A_k p_k = -\widehat{c}_k.$$

We know from the optimality conditions for (3.11.1) that its Lagrange multiplier vector μ_k must satisfy the following two conditions:

$$g_k + B_k p_k = A_k^T \mu_k \tag{3.11.2a}$$

$$\mu_k \geq 0. \tag{3.11.2b}$$

Comparing the conditions of Theorem 3.8.4 and (3.11.2), we see that as $p_k \to 0$, the Lagrange multipliers of the subproblem (3.11.1) approach the Lagrange multipliers λ^* of NIP. In fact, it can be shown that if $A(x^*)$ has full rank and x^* satisfies the sufficient conditions of Theorem 3.8.4, then the QP subproblem (3.11.1) *will identify the correct active set* if x_k is "sufficiently close" to x^* (see [Robinson 1974]). This favorable result suggests that the active set of the QP subproblem can be taken as a prediction of the active set of NIP. (However, the working set is "implicit", in the sense that it is obtained as the solution of a subproblem.) Enormous algorithmic advantage can be taken of this property if the subproblem (3.11.1) is solved using a QP method that permits a "warm start" (i.e., that can exploit a prediction of the active set to enhance efficiency).

The treatment of inequality constraints in the definition of a merit function for SQP methods has been approached in several ways, of which we mention two. The ℓ_1 penalty function for inequalities has been the most common choice (see [Han 1976], [Powell 1978]), because of the desirable properties mentioned in Section 3.5.4. However, recent work suggests that an augmented Lagrangian merit function can also be extended to inequality constraints. Schittkowski [1981] and Gill, Murray, Saunders and Wright [1986] have proved global convergence of an SQP method based on an augmented Lagrangian function. The latter algorithm performs a linesearch not only with respect to x and λ (as in Section 3.5.4), but also with respect to a set of nonnegative *slack variables*. The slack variables are introduced *purely during the linesearch* to convert the inequality constraint $c_i(x) \geq 0$ to an equality constraint:

$$c_i(x) \geq 0 \qquad \text{if and only if} \qquad c_i(x) - s_i = 0, \quad s_i \geq 0.$$

The merit function then has the form

$$M_A(x, \mu, s, \rho) \equiv f(x) - \mu^T(c(x) - s) + \tfrac{1}{2}\rho(c(x) - s)^T(c(x) - s),$$

where μ is an estimate of the multiplier vector (see (3.8.5)).

3.12. Sequential Linearly Constrained Methods

Exactly as in extending an SQP method to treat inequality constraints, the obvious strategy in an SLC method (see Section 3.6) is to formulate the subproblem with

linearized *inequality* constraints. A typical subproblem in an SLC method will thus have the form:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \mathcal{F}_k(x) \\ \text{subject to} & \mathcal{A}_k(x - x_k) \geq -c_k, \end{array}$$

where $\mathcal{F}_k(x)$ is (as before) a general approximation to the Lagrangian function at x_k, based on a current multiplier estimate λ_k.

For this SLC method, it can be shown [Robinson 1972] that the subproblem will identify the correct active set in a sufficiently small neighborhood of the solution, and that the favorable local convergence rate is therefore maintained (see Section 3.6). An SLC method (the code MINOS; [Murtagh and Saunders 1983]) is widely viewed as the most effective algorithm for solving *large-scale* problems of the form NIP.

When $\mathcal{F}_k(x)$ is defined as an augmented Lagrangian function (see (3.7.1)), some decision must be made about which constraints are to be treated as active. Any strategies designed to permit early termination of unpromising subproblems must exercise special caution to ensure that the multiplier estimates have the properties needed to achieve global convergence.

3.13. Augmented Lagrangian Methods

The application of augmented Lagrangian methods to problems with inequalities is much less straightforward than for SQP and SLC methods (in which the constraints in the subproblem become inequalities), or for penalty function methods (in which the penalty term includes the violated inequality constraints). Augmented Lagrangian methods do *not* include constraints in the associated subproblem, and furthermore do not approach x^* via a sequence of points where the active constraints are violated. Hence, some other strategy must be devised to identify the active constraints.

Once a "working set" $\widehat{c}(x)$ of constraints has been identified, an augmented Lagrangian function can be defined that includes only the working set:

$$L_A(x, \lambda, \rho) = f(x) - \lambda^T \widehat{c}(x) + \tfrac{1}{2}\rho \widehat{c}(x)^T \widehat{c}(x).$$

However, tests must be made at every stage to ensure that the working set is correct, and great difficulties have been encountered in developing a general set of guidelines for this purpose. Other approaches are based on enforcing sign restrictions on the Lagrange multiplier estimates (see, e.g., [Rockafellar 1973]).

3.14. Barrier-Function Methods

In this final section, we turn to a class of methods with a different flavor from any suggested previously for solving NIP. *Barrier-function methods* can be applied only to inequality constraints for which a strictly feasible initial point exists. Thereafter, a barrier-function method generates a sequence of strictly feasible iterates. These methods have received enormous attention recently because of their close relationship with the "new" polynomial approaches to linear programming (see Chapter 2 (linear programming), [Karmarkar 1984], and [Gill, Murray, Saunders, Tomlin and Wright 1986]).

3.14.1. Motivation

In many physical and engineering applications, the constraint functions not only characterize the desired properties of the solution, but also define a region in which the problem statement is meaningful (for example, $f(x)$ or some of the constraint functions may be undefined outside the feasible region). An artificial convention for extending the problem statement outside the feasible region would not lend itself to the design of a computationally reasonable algorithm, and might introduce complications not present in the original problem.

Barrier-function methods require strict satisfaction of all constraints at the starting point and subsequent iterates. The continued enforcement of feasibility is the "opposite" of a penalty function method for inequalities (Section 3.10), where the constrained solution is approached through a sequence of strictly *infeasible* points with respect to the active constraints.

As in the penalty case, a barrier-function method creates a sequence of modified functions whose successive unconstrained minimizers should converge in the limit to the constrained solution. In general, the unconstrained minimizer of f will be infeasible, or f may be unbounded below. In order to guarantee that successive iterates are feasible, the modified objective function includes a term to keep iterates "inside" the feasible region. If a "barrier" is created at the boundary of the feasible region by constructing a continuous function with a positive singularity, any unconstrained minimizer of the modified function must lie strictly inside the feasible region. If the weight assigned to the barrier term is decreased toward zero, the sequence of unconstrained minimizers should generate a strictly feasible approach to the constrained minimizer.

The two most popular barrier functions are the *logarithmic* barrier function, usually attributed to Frisch [1955]:

$$B(x,r) = f(x) - r\sum_{i=1}^{m_N} \ln(c_i(x)), \tag{3.14.1}$$

and the inverse barrier function [Carroll 1961]:

$$B(x,r) = f(x) + r\sum_{i=1}^{m_N} \frac{1}{c_i(x)}. \tag{3.14.2}$$

The positive weight r in (3.14.1) and (3.14.2) is called the *barrier parameter*.

We henceforth consider only the logarithmic barrier function (3.14.1). Fiacco and McCormick [1968] present a convergence proof that, under quite general conditions on f and $\{c_i\}$, there exists a compact set containing x^* within which the sequence $\{x^*(r)\}$, the minimizers of successive $B(x,r)$, converges to x^* as $r \to 0$. As in the analogous proof for penalty function methods, the conditions for convergence do not require satisfaction of the constraint qualification at the limit point, so that barrier-function methods will converge to minimizers not satisfying the Karush-Kuhn-Tucker conditions. However, the proof of convergence requires that x^* must

lie in the closure of the interior of the feasible region, and consequently x^* is not permitted to be isolated from strictly feasible points.

Because convergence of $x^*(r)$ to x^* is guaranteed only within a compact set including x^*, it is possible for the logarithmic barrier function to be unbounded below, as in an example given by Powell [1972]:

$$\underset{x \in \mathbb{R}^1}{\text{minimize}} \quad -\frac{1}{x^2+1} \quad \text{subject to} \quad x \geq 1.$$

The solution is $x^* = 1$, but the logarithmic barrier function is given by

$$B(x,r) = -\frac{1}{x^2+1} - r \ln(x-1),$$

which is unbounded below for any $r > 0$. However, unboundedness is much less likely to happen than with penalty functions, because the feasible region is often compact.

3.14.2. Properties

Let $\{r_k\}$ be a strictly decreasing positive sequence, with $\lim_{k\to\infty} r_k = 0$. The minimizers of successive barrier functions exhibit the following properties, where B_k denotes $B(x^*(r_k))$, f_k denotes $f(x^*(r_k))$ and c_k denotes $c(x^*(r_k))$:

1. $\{B_k\}$ is strictly decreasing for sufficiently small r_k and bounded c_k;
2. $\{f_k\}$ is nonincreasing;
3. $-\sum_{i=1}^{m_N} \ln c_i(x^*(r_k))$ is nondecreasing.

Property (iii) does *not* imply that the constraint values decrease at successive $x^*(r_k)$. A reduction in the barrier parameter allows the constraints to approach the boundary of the feasible region, but does not enforce a decrease.

By definition of an unconstrained minimizer, the following relation holds at $x^*(r)$:

$$\nabla B = g - r \sum_{i=1}^{m_N} \frac{1}{c_i} a_i = g - \mathcal{A}^T \begin{pmatrix} r/c_1 \\ \vdots \\ r/c_{m_N} \end{pmatrix} = 0. \tag{3.14.3}$$

Since $r > 0$ and $c_i > 0$ for all i, (3.14.3) shows that the gradient of f at $x^*(r)$ is a *nonnegative linear combination* of *all* the constraint gradients, where the coefficient of a_i is r/c_i. As r approaches zero, the quantity $r/c_i(x^*(r))$ will converge to zero if c_i is not active at x^*, since c_i is strictly bounded away from zero in a neighborhood of x^*. Assume that m constraints are active at x^*. Then for sufficiently small r, the relation holding at $x^*(r)$ can be written:

$$g = A^T \begin{pmatrix} r/\hat{c}_1 \\ \vdots \\ r/\hat{c}_m \end{pmatrix} + O(r), \tag{3.14.4}$$

where $\widehat{c}_i$ denotes the i-th active constraint, and A denotes the $m \times n$ matrix of active constraint gradients.

It follows from (3.14.4) that the quantity

$$\lambda_i(r) \equiv \frac{r}{\widehat{c}_i(x^*(r))}, \tag{3.14.5}$$

defined only for the active constraints, satisfies a relationship with g and A analogous to the multiplier relation that must hold at x^* if $A(x^*)$ has full rank. The vector $\lambda(r)$ satisfies $\lambda(r) = \lambda^* + O(r)$, where λ^* is the vector of Lagrange multipliers at x^*.

As with the quadratic penalty function (Section 3.3), the barrier parameter can be considered as an independent variable defining a trajectory of values $x(r)$. The following result is proved in Fiacco and McCormick [1968].

Theorem 3.14.1. *If $A(x^*)$ has full rank, and the sufficient conditions of Theorem 3.8.4 hold at x^*, then for sufficiently small r, there exists a continuously differentiable trajectory $x(r)$ such that*

$$\lim_{r \to 0} x(r) = x^*,$$

and for any $r > 0$, $x(r)$ is a local minimizer of $B(x, r)$. ∎

The trajectory $x(r)$ has several interesting properties. Expanding about $r = 0$ gives the following expression:

$$x(r) = x^* + ry + O(r^2),$$

where

$$y = \lim_{r \to 0} \frac{x(r) - x^*}{r} = \left. \frac{dx(r)}{dr} \right|_{r=0}. \tag{3.14.6}$$

Differentiating the identity (3.14.3) and using (3.14.6), we obtain the following expression (*cf.* (3.3.9) for the penalty case):

$$Ay = \begin{pmatrix} 1/\lambda_1^* \\ \vdots \\ 1/\lambda_m^* \end{pmatrix}. \tag{3.14.7}$$

The relationship (3.14.7) implies that the minimizers of successive barrier functions do not approach x^* tangentially to any constraint for which $0 < \lambda_i^* < \infty$.

Barrier-function methods are not well suited to application of general-purpose unconstrained minimization techniques, primarily because of ill-conditioning of the Hessian matrices at $x^*(r)$ if $0 < m < n$. The ill-conditioning of the Hessian matrices of barrier functions does not result from the influence of the barrier parameter,

but rather from the singularities caused by the active constraints. The Hessian of $B(x^*(r), r)$ is given by:

$$H - \sum_{i=1}^{m_N} \frac{r}{c_i} H_i + \mathcal{A}^T \begin{pmatrix} r/c_1^2 & & \\ & \ddots & \\ & & r/c_{m_N}^2 \end{pmatrix} \mathcal{A}, \tag{3.14.8}$$

where all quantities are evaluated at $x^*(r)$. For inactive constraints, c_i is bounded away from zero, and thus the quantities r/c_i and r/c_i^2 go to zero as $x^*(r)$ approaches x^*. For the active constraints, we know from (3.14.5) that $r/c_i(x^*(r))$ approaches the corresponding (bounded) Lagrange multiplier. For a nonzero Lagrange multiplier, the ratio $r/c_i(x^*(r))^2$ is thus unbounded as $c_i \to 0$. The first two terms of (3.14.8) constitute an increasingly accurate approximation to $\nabla^2 L(x^*, \lambda^*)$, the Hessian of the Lagrangian function at the solution. However, if $0 < m < n$, or $m = n$ and A has some dependent rows, the dominant rank-deficient matrix causes the condition number of (3.14.8) to become unbounded. In particular, $\nabla^2 B(x^*(r), r)$ has m unbounded eigenvalues as $r \to 0$, with corresponding eigenvectors in the range of $A(x^*)^T$. The remaining $(n - m)$ eigenvalues are bounded and their eigenvectors lie in the null space of $A(x^*)$. (See [Murray 1971b], for details.)

References

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

[2] J. R. Bunch and L. C. Kaufman [1980]. A computational method for the indefinite quadratic programming problem, *Linear Algebra and its Applications* 34, 341–370.

[3] C. W. Carroll [1961]. The created response surface technique for optimizing nonlinear restrained systems, *Operations Research* 9 , 169–184.

[4] R. W. Chamberlain, C. Lemaréchal, H. Pedersen and M. J. D. Powell [1982]. The watchdog technique for forcing convergence in algorithms for constrained optimization, *Mathematical Programming Study* 16, 1–17.

[5] T. F. Coleman and A. R. Conn [1982a]. Nonlinear programming via an exact penalty function: asymptotic analysis, *Mathematical Programming* 24, 123–136.

[6] T. F. Coleman and A. R. Conn [1982b]. Nonlinear programming via an exact penalty function: global analysis, *Mathematical Programming* 24, 137–161.

[7] R. Courant [1943]. Variational methods for the solution of problems of equilibrium and vibrations, *Bulletin of the American Mathematical Society* 49, 1–23.

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

[9] J. J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart [1979]. *LINPACK Users Guide*, SIAM Publications, Philadelphia.

[10] I. S. Duff and J. K. Reid [1983]. The multifrontal solution of indefinite sparse symmetric linear equations, *ACM Transactions on Mathematical Software* 9, 302–325.

[11] A. V. Fiacco and G. P. McCormick [1968]. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York and Toronto.

[12] R. Fletcher [1974]. Methods related to Lagrangian functions, in P. E. Gill and W. Murray (eds.), *Numerical Methods for Constrained Optimization*, Academic Press, London and New York, 219–240.

[13] R. Fletcher [1981]. *Practical Methods of Optimization, Volume 2, Constrained Optimization*, John Wiley and Sons, New York and Toronto.

[14] R. Fletcher [1985]. An ℓ_1 penalty method for nonlinear constraints, in P. T. Boggs, R. H. Byrd and R. B. Schnabel (eds.), *Numerical Optimization 1984*, SIAM, Philadelphia, 26–40.

[15] R. Fletcher [1986]. Recent developments in linear and quadratic programming, Report NA94, Department of Mathematical Sciences, University of Dundee, Scotland.

[16] K. R. Frisch [1955]. The logarithmic potential method of convex programming, Memorandum of May 13, 1955, University Institute of Economics, Oslo, Norway.

[17] P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders and M. H. Wright [1984]. A weighted Gram-Schmidt method for convex quadratic programming, *Mathematical Programming* 30, 176–195.

[18] P. E. Gill and W. Murray [1974]. Newton-type methods for unconstrained and linearly constrained optimization, *Mathematical Programming* 28, 311–350.

[19] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright [1986]. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, *Mathematical Programming* 36, 183–209.

[20] P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright [1984]. Sparse matrix methods in optimization, *SIAM J. on Scientific and Statistical Computing* 5, 562–589.

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

[22] P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright [1987]. A Schur-Complement method for sparse quadratic programming, Report SOL 87-12, Department of Operations Research, Stanford University.

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

[24] D. Goldfarb and A. Idnani [1983]. A numerically stable dual method for solving strictly convex quadratic programs, *Mathematical Programming* 27, 1–33.

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

[26] N. I. M. Gould [1986]. On the accurate determination of search directions for simple differentiable penalty functions, *Institute of Mathematics and its Applications J. Numerical Analysis* 6, 357–372.

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

[28] S.-P. Han [1977]. A globally convergent method for nonlinear programming, *J. Optimization Theory and Applications* 22, 297–310.

[29] M. R. Hestenes [1969]. Multiplier and gradient methods, *J. Optimization Theory and Applications* 4, 303–320.

[30] N. Karmarkar [1984]. A new polynomial-time algorithm for linear programming, *Combinatorica* 4, 373–395.

[31] H. W. Kuhn and A. W. Tucker [1951]. "Nonlinear Programming", in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (J. Neyman, ed.), 481–492, Berkeley, University of California Press.

[32] F. A. Lootsma [1969]. Hessian matrices of penalty functions for solving constrained optimization problems, *Philips Res. Repts* **24**, 322–331.

[33] D. G. Luenberger [1984]. *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Menlo Park, California.

[34] N. Maratos [1978]. *Exact Penalty Function Algorithms for Finite-Dimensional and Control Optimization Problems*, Ph. D. Thesis, University of London.

[35] W. Murray [1969]. An algorithm for constrained minimization, in R. Fletcher (ed.), *Optimization*, Academic Press, London and New York, 247–258.

[36] W. Murray [1971a]. An algorithm for finding a local minimum of an indefinite quadratic program, Report NAC 1, National Physical Laboratory, England.

[37] W. Murray [1971b]. Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions, *J. Optimization Theory and Applications* 7, 189–196.

[38] B. A. Murtagh and M. A. Saunders [1982]. A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints, *Mathematical Programming Study* 16, 84–117.

[39] B. A. Murtagh and M. A. Saunders [1983]. MINOS 5.0 User's Guide, Report SOL 83-20, Department of Operations Research, Stanford University.

[40] J. M. Ortega and W. C. Rheinboldt [1970]. *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, London and New York.

[41] M. J. D. Powell [1969]. A method for nonlinear constraints in minimization problems, in R. Fletcher (ed.), *Optimization*, Academic Press, London and New York, 283–298.

[42] M. J. D. Powell [1972]. Problems relating to unconstrained optimization, W. Murray (ed.), *Numerical Methods for Constrained Optimization*, Academic Press, London and New York, 29–55.

[43] M. J. D. Powell [1978]. The convergence of variable metric methods for nonlinearly constrained optimization calculations, in O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), *Nonlinear Programming 3*, Academic Press, London and New York, 27–63.

[44] S. M. Robinson [1972]. A quadratically convergent algorithm for general nonlinear programming problems, *Mathematical Programming* 3, 145–156.

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

[46] R. T. Rockafellar [1973]. The multiplier method of Hestenes and Powell applied to convex programming, *J. Optimization Theory and Applications* 12, 555–562.

[47] J. B. Rosen [1978]. Two-phase algorithm for nonlinear constraint problems, in O. L. Mangasarian, R. R. Meyer, and S. M. Robinson (eds.), *Nonlinear Programming 3*, Academic Press, London and New York, 97–124.

[48] J. B. Rosen and J. Kreuser [1972]. A gradient projection algorithm for nonlinear constraints, in F. A. Lootsma (ed.), *Numerical Methods for Non-Linear Optimization*, Academic Press, London and New York, 297–300.

[49] K. Schittkowski [1981]. The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function, *Numerische Mathematik* 38, 83–114.

[50] R. A. Tapia [1977]. Diagonalized multiplier methods and quasi-Newton methods for constrained optimization, *J. Optimization Theory and Applications* 22, 135–194.

[51] M. H. Wright [1976]. *Numerical Methods for Nonlinearly Constrained Optimization*, Ph. D. Thesis, Stanford University, California.