Numerical Methods for Nonlinearly Constrained Optimization

Margart Ann Hagen Wright

SLAC-R-193

Prepared for the Department of Energy
under contract number DE-AC02-76SF00515

Printed in the United States of America. Available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

NUMERICAL METHODS FOR NONLINEARLY CONSTRAINED OPTIMIZATION

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Margaret Ann Hagen Wright

March 1976

ACKNOWLEDGMENTS

It is difficult to convey all the contributions that Dr. Walter Murray has made to this dissertation. He has been an unfailing source of algorithmic and computational insights, displayed extreme generosity with his time and ideas, and generated a contagious enthusiasm for all aspects of this work. In addition to technical guidance, he has provided a constant friendship; my association with him is, for me, a rare privilege.

I also wish to acknowledge help from the following:

- Professor Gene H. Golub, for introducing me to numerical linear algebra and computer programming, for encouraging me to begin work on a Ph.D., and for financial support throughout;
- The Division of Numerical Analysis and Computing at the National Physical Laboratory, England, and, in particular, the Optimization Group, for the valuable and rewarding seven months I spent there;
- Professor Forest Baskett, Dr. Jerome Friedman, end the Computation Research Group at the Stanford Linear Accelerator Center, for providing the stimulating environment of computer science research in which my later work has been carried out;
- Dr. Philip Gill, for many helpful discussions of details of optimization theory and practice;
- Professor Joseph Oliger, for serving on my reading committee;
- Mrs. Harriet Canfield, for her patience and meticulous typing, even through the numerous complicated equations.

I give personal thanks to Eric, for his continuing inspiration and encouragement; and to Amy, who has often wondered why so much work was needed to acquire three letters.

* * * *

This research was supported by the following institutions:

National Science Foundation Grant DCR71-01996 A04-05;

National Science Foundation Grant DCR7203712, A2;

Energy Research-and Development Administraticn under contract AT(04-3) 326 PA #30;

Energy Research and Development Administration under contract E(043)515.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page No.

INTRODUCTION

This dissertation presents a detailed derivation and description of a new feasible-point algorithm for nonlinearly constrained optimization. The new method is based on the properties of the trajectory of minima obtained by varying the barrier parameter of the logarithmic barrier function. The algorithm is closely related to a nonfeasible-point method based on the trajectory of the quadratic penalty function, due to Walter Murray, and this method is also described in detail.

A description is given of penalty and barrier function methods, and of some methods which, like the trajectory methods, are based on the Lagrangian function. This discussion is included in order to facilitate theoretical and numerical comparison of the trajectory methods with other algorithms. It also serves to clarify the motivation for the various methods, since nearly all the algorithms discussed have been developed to overcome the deficiencies of penalty and barrier function methods.

A comprehensive set of programs has been de-$eloped, implementing the viable methods discussed, and a selection of typical numerical results is included. A primary concern in this dissertation has been with practical algorithms, and the need to consider methods that will converge even *from* a poor initial estimate of the solution. Many "sophisticated" algorithms often critically depend on properties that hold in a close neighborhood of the solution and, consequently, may perform poorly or even fail. for any initial point not in such a region. The experimentation with the methods has, therefore, been directed toward analyzing their general behavior, as well as asymptotic rates of convergence.

To the best of my knowledge, the following results are original:

(1) The special linear searches for the logarithmic barrier function (Chapter 3);

(2) All aspects of the barrier trajectory algorithm (half of Chapter 5);

(3) The uniform implementation and testing of algorithms (Chapter 6).

The portions of Chapter 5 that deal with the penalty trajectory algorithm are based on the work of Walter Murray, but the algorithm has not previously been presented in the form given here.

Definition of Symbols

The notation of this dissertation corresponds in most cases to that given in P. Gill and W. Murray (eds.), Numerical Methods for Constrained Optimization, Academic Press, 1974.

The following definitions are used throughout:

x - a real vector of dimension n, given by

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \\ \\ x_n \end{bmatrix} .$$

$|| \; ||$ - The Euclidean norm of a vector, i.e., $||x|| = \sqrt{x_1^2 + \ldots + x_n^2}$; or the induced matrix norm.

$F(x)$ - The scalar function of n variables whose minimum is to be found.

$g(x)$ - The gradient vector of $F(x)$, whose ith component is given by $g_i(x) = \dfrac{\partial F(x)}{\partial x_i}$.

$G(x)$ - The n by n Hessian matrix of $F(x)$, whose (i,j)th element is given by $\dfrac{\partial^2 F(x)}{\partial x_i \partial x_j}$.

$c_i(x)$ - The ith constraint function; the set of constraints is sometimes represented by the vector $c(x)$, whose ith component is given by c_i.

$a_i(x)$ - The gradient vector of $c_i(x)$.

$A(x)$ - A matrix whose ith column is $a_i(x)$, the gradient of $c_i(x)$.

$G_i(x)$ - The n by n Hessian matrix of $e_i(x)$.

x^* - The solution of the currently considered constrained optimization problem.

λ^* - The vector of Lagrange multipliers at x^* (defined in Section 1.2).

A - A vector that normally denotes an estimate of A*.

Lagrangian
Function - The function $F(x) - \lambda^{*T}c(x)$.

The following two matrices are defined in terms of an n by m matrix A, of rank r:

Q_1 - An n by r matrix whose columns form an orthogonal basis for the space spanned by the columns of A.

Z - An n by (n-r) matrix whose columns form an orthogonal basis for the null space of the columns of A.

CHAPTER 1. Theoretical Background

CHAPTER 1

THEORETICAL BACKGROUND

1.1 Statement of Problem

.This dissertaticn is concerned with methods for the numerical solution of the following problems:

P1 - Equality Constrained Problem

$$\underset{x}{\text{minimize}}\ F(x),\ x\in E^n$$

subject to the constraints

$$c_i(x) = 0, \quad i = 1,2,\ldots,m;$$

P2 - Inequality Constrained Problem

$$\underset{x}{\text{minimize}}\ F(x),\ x\in E^n$$

subject to the constraints

$$c_i(x) \geq 0, \quad i = 1,2,\ldots,\ell.$$

In P1 and P2, $F(x)$ and $\{c_i(x)\}$ are scalar functions of the n independent variables $\{x_1,x_2,\ldots,x_n\}$. The function $F(x)$ is often called the "objective function", and $\{c_i(x)\}$ are the "constraint functions". F and $\{c_i\}$ will be assumed to have at least continuous first and second derivatives.

A further problem could be considered, where some of the constraint functions are equality constraints and others are inequalities, but the notation for such a problem becomes more complicated. Ail subsequent discussion will. concern either problem P1 or P2, and it should be clear how the separately developed results would apply to a mixed problem.

Definition 1: A point $\hat{x}$ is feasible with respect to the equality constraint $c_i(x) = 0$ if $c_i(\hat{x}) = 0$, and infeasible if $c_i(\hat{x}) \neq 0$. The constraint $c_i(x)$ is satisfied at a feasible point $\hat{x}$. Given a set of equality constraints $\{c_i(x) = 0,\ i = 1,2,\ldots,m\}$, $\hat{x}$ is feasible with respect to the set of constraints if $c_i(\hat{x}) = 0$, for all $i = 1,2,\ldots,m$; $\hat{x}$ is infeasible with respect to the set of constraints if $c_i(\hat{x}) \neq 0$ for any $i \in \{1,2,\ldots,m\}$.

Definition 2: A point $\hat{x}$ is feasible with respect to the inequality constraint $c_j(x) \geq 0$ if $c_j(\hat{x}) \geq 0$, strictly feasible if $c_j(\hat{x}) > 0$, and infeasible *if* $c_j(\hat{x}) < 0$. If $c_j(\hat{x}) \geq 0$, the constraint c_j is satisfied at $\hat{x}$, and c_j is strictly satisfied if $c_j(\hat{x}) > 0$. The definition of feasibility with respect to a set of inequality constraints is similar to the equality case.

We shall be concerned only with local minima, defined as follows:

Definition 3: The point x^* is a ~~weak~~ ~~local~~ minimm of problem P1 or P2 if:

(1) x^* is feasible with respect to all the problem constraints;

(2) there exists $6 > 0$ such that $F(x^*) \leq F(x)$ for all x satisfying:

(a) $||x - x^*|| \leq \delta$; and

(b) x is feasible with respect to all the problem ccnstraints.

If $F(x^*) < F(x)$ for similarly defined x, $x \neq x^*$, then x^* is a ~~strong~~ local minimum of P1 or P2. When the term "local minimum" is used, it should be interpreted to mean “strong local minimum” unless otherwise specified.

1.2 Conditions for a Minimum

Algorithms to solve problem P1 or P2 are based on trying to identify a point satisfying the conditions for a local minimum. In this section one form of these conditions will be derived; we shall not be concerned with stating the weakest possible conditions for a local minimum, since we are interested only in properties thet can be computationally verified.

1.2.1 Unconstrained Minimum

If there are no constraints in problem P1 or P2, the minimization problem reduces to:

P3 - Unconstrained Problem

$$\min_{x} \text{minimize } F(x),\ x\epsilon E^n .$$

Before studying the more general problems P1 and E, we will derive necessary and sufficient conditions that hold at a local minimum of P3. Since F(x)∈C2, it may be expanded about a point x* in a Taylor series:

$$F(x^* + \epsilon h) = F(x^*) + \epsilon hTg(x^*) + \frac{1}{2} E2\, hTG(x^*)h + O(|\epsilon|^3), \qquad (1.2.1)$$

where $||h|| = 1$

Condition 1: A necessary condition for a point x* to be a local minimum of the unconstrained problem P3 is that $||g(x^*)|| = G.$

If $||g(x^*)||$ is not zero, then there exists some direction h for which $h^T g(x^*) < 0$. Hence, from (1.2.1) there exists a positive scalar E' > 0 such that for any ε, 0 < E < E',

$$F(x^* + \epsilon h) < F(x^*),$$

and x* cannot be a local- minimum.

Condition 2: A necessary condition for a point x* to be a weak local minimum of P3 is that G(x*) be positive semi-definite.

If $||g(x^*)|| = 0$, then from (1.2.1),

$$F(x^* + \epsilon h) = F(x^*) + \frac{1}{2} \epsilon 2hTG(x^*)h + O(|\epsilon| 3). \qquad (1.2.2)$$

If the matrix G(x*) is indefinite, there exists a unit vector h such that $h^T G(x^*)h < 0$. Hence, from (1.2.2) there exists a positive scalar E' such that for any E, 0 < |ε| < E',

$$F(x^* + \epsilon h) < F(x^*),$$

contradicting that x* is a local minimum.

Condition 3: Sufficient conditions for x^* to be a strong local minimum of P3 are:

(1) $||g(x^*)|| = 0$;

(2) $G(x^*)$ is positive definite.

The condition $||g(x^*)|| = 0$ is necessary for x^* to be a local minimum. If $G(x^*)$ is positive definite, $h^TG(x^*)h > 0$ for all non-zero vectors h. From (1.2.2) it follows that there exists $\epsilon' > 0$ such that for any ϵ satisfying $0 < |\epsilon| < \epsilon'$

$$F(x^*) < F(x^* + \epsilon h),$$

for any vector h, $||h|| = 1$. Therefore, there exists a neighborhood of x^* such that the smallest value of F for any point in the neighborhood occurs at x^*, and x^* is a strong local minimm.

1.2.2 Equality Constraints

In deriving necessary and sufficient conditions for a point x^* to be a local minimum of the equality constrained problem P1, the definition of a local minimum indicates that we need to consider the behavior of $F(x)$ only along feasible perturbations from x^*. If a characterization of a feasible move from x^* is possible, the Taylor series expansion of F about x^* along such a perturbation can be examined as in the unconstrained case.

The following initial results develop the existence of a relationship between the gradients of $F(x)$ and $\{c_i(x)\}$ at a point.

Farkas' Lemma: Given a set of n-vectors. $\{a_i\}$, $i = 1,2,\ldots,m$, and a vector b, a necessary and sufficient condition for the existence of non-negative scalar values $\{\lambda_i\}$, $i = 1,2,\ldots,m$, such that

$$b = \sum_{i=1}^{m} \lambda_i a_i, \quad \lambda_i \geq 0,$$

is that for every vector y such that $a_i^Ty \geq 0$, $i=1,2,\ldots,m$, it follows that $b^Ty \geq 0$.

An alternative statement of Farkas' Lemma is that there exists a vector z such that $a_i^T z \geq 0$, $i = 1,2,\ldots,m$, and $b^T z < 0$, if and only if there is no vector λ satisfying $b = \sum_{i=1}^{m} \lambda_i a_i$, $\lambda_i \geq 0$.

Theorem 1 (Existence of Lagrange Multipliers; Kuhn-Tucker conditions):

If:

(a) F and $\{c_i\}$, $i = 1,2,\ldots,m$ are once differentiable;

(b) $c_i(\hat{x}) = 0$, $i = 1,2,\ldots,m$;

(c) at $\hat{x}$, for all vectors p such that $A(\hat{x})^T p = 0$, where the columns of A are $\{a_i\}$, it holds that $g(\hat{x})^T p \geq 0$, then:

There exists a vector λ, of length m, such that $g = \sum_{i=1}^{m} \lambda_i a_i$, or, equivalently, $g = A\lambda$.

If the conditions of Theorem 1 hold, the point $\hat{x}$ is said to satisfy the first-crier Kuhn-Tucker conditions.

Proof: In the following discussion, all vector and matrix functions are evaluated at $\hat{x}$ unless otherwise specified. Note that if no vector p exists such that $A^T p = 0$, the theorem is vacuously satisfied.

If a vector p exists such that $A^T p = 0$, i.e., $A^T p \geq 0$ and $-A^T p \geq 0$, then by assumption it also holds that $g^T p \geq 0$. Applying Farkas' Lemma to the set of 2m vectors $\{a_i\}$, $\{-a_i\}$, and the vector g, there must exist a vector u, of length 2m, such that

$$g = [A \;\vdots\; -A]u, \quad u_i > 0,$$

or, letting $\lambda_i = u_i - u_{i+m}$,

$$g = Ah .$$

■

An alternative statement of this result is that if there exists no vector p simultaneously satisfying $A^T p = 0$ and $g^T p < 0$, then the vector g must lie entirely in the range of A.

In order to use Theorem 1 to describe conditions at a local minimum of P1, restrictions on the constraint functions are required.

First-Order Constraint Qualification (Kuhn and Tucker, 1951; Fiacco and McCormick, 1968):

The first-order constraint qualification holds at a point $\hat{x}$ if:

(1) $\{c_i\}$, $i = 1,2,\ldots,m$, are once continuously differentiable;

(2) $c_i(\hat{x}) = 0$, for all $i = 1,2,\ldots,m$;

(3) every non-zero vector p satisfying:

$A(\hat{x})^T p = 0$ is tangent to a once-differentiable arc emanating from $\hat{x}$, along which all constraints are satisfied on the arc in some neighborhood of $\hat{x}$ An arc is a directed differenxiable curve in E^n parameterized by a single parameter 8, $0 \leq \theta \leq E$.

In the following discussion, all vector and matrix Tunctions are evaluated at $\hat{x}$ unless otherwise specified. Any possible feasible move from $\hat{x}$ must be along a direction p satisfying $A^T p = 0$, since

$$c(\hat{x} + \epsilon p) = c(\hat{x}) + \epsilon A^T p + O(|\epsilon|^2), \text{ for } ||p|| = 1.$$

Such a vector p may be written as Zp_N where Z is a matrix whose columns form an orthogcnai basis for the null space of A, so that $A^T Z = 0$, $Z^T Z = I$. Geometrically, the first-order constraint qualification implies that every direction that makes no first-order change in any Constraint at $\hat{x}$ must be the beginning of a feasible arc, or, equivalently, that a sufficiently small step taken along such a direction must be arbitrarily close to a feasible point.

The constraint qualification enables all possible feasible perturbations from $\hat{x}$ to be characterized, since, if it is satisfied, any direction $p = Zp_N$ is the beginning of a feasible path. If the constraint qualification does not hold at a point, then there can . exist airections $p = Zp_N$ along which no feasible move is possible.

We are now able to prove the following result:

Theorem 2 (First-Order Necessary Conditions for a Minimum of P1)

If:

(1) F and $\{c_i\}$, $i = 1,2,\ldots,m$, are differentiable at X^*; and

(2) the first-order constraint qualification holds at x^*, then:

A necessary condition for x^* to be a local minimum of P1 is that there exist a vector λ^*, of length m, such that $g(x^*) = A(x^*)\lambda^*$, or, equivalently, $g(x^*)$ is in the range of $A(x^*)$.

Proof: In the following discussion, all vector and matrix functions ere evaluated at x^* unless otherwise specified.

Assume that x^* is a local minimum of P1. Consider all vectors p such that $A^Tp = 0$. If there are none, then assuredly g is in the range of A, since the columns of A must span the entire space.

If there are such vectors p, then, because the constraint qualification holds at x^*, p is the tangent of a once-differentisble feasible arc, emanating from x^*. Because x^* is a local minimum, F(x) must not decrease along the arc from x^*, since all points on the arc are feasible in some neighborhood of x^*, and hence $g^Tp \geq 0$. Therefore, the conditions of Theorem 1 are satisfied at x', and Q lies in the range of A. The vector $\lambda^*$is called the vector of Lagrange multipliers.

Note the importance of the constraint qualification in this proof. A point x^* is a local minimum if $g^Tp \geq 0$ for every feasible perturbation p. The condition that $A^T p = 0$, which is necessary for p to be a feasible move from x^*, is made a sufficient condition by the constraint qualification.

The verificaticn of the constraint qualification at a possible minimum x^* is not computationally convenient. A mcre practical condition, which implies that the constraint qualification holds, is that the matrix $A(x^*)$ be of full rank, i.e., the gradients of the constraints are linearly independent.

Theorem 3. If:

(1) $\{c_i\}$, $i = 1,2,\ldots,m$, are continuously differentiable;

(2) $c_i(x^*) = 0$, $i = 1,2,\ldots,m$; and

(3) $A(x^*)$ is of full rank,

then the first-order constraint qualification holds at x^*.

Proof: Let p satisfy $A(x^*)^T p = 0$. If there is no such p, the constraint qualification holds vacuously.

Otherwise, p must be expressible in the form Zp_N for some vector p_N. Construct the arc $\alpha(\theta)$ from x^* by specifying $\alpha(0) = x^*$, and $\frac{d}{d\theta}(\alpha(\theta)) = Z(\alpha(\theta))p_N$, for $0 \leq \theta \leq \epsilon$, where $Z(\alpha(\theta))$ denotes a particular representation of a matrix whose columns form an orthogonal basis for the null space of $A(\alpha(6))$. Because the columns of A are linearly independent at x^*, and the functions $\{a_i\}$ are continuous, the matrix $A(\alpha(\theta))$ is continuous and of full rank in a neighborhood of x^*; thus an appropriate representation of the matrix $Z(\alpha(\theta))$ is also continuous and of fixed rank in a neighborhood of x^*, and so the given definition of $\alpha(\theta)$ specifies a continuous, differentiable arc.

Expanding c_i along the arc:

$$c_i(\alpha(\bar\theta)) = c_i(\alpha(0)) + \bar\theta \frac{\partial c_i}{\partial \theta}(\alpha(\eta))\Bigg|_{0 \le \eta \le \bar\theta}$$

$$= c_i(\alpha(0)) + \bar\theta a_i^T(\alpha(\eta)) \frac{d}{d\theta}(\alpha(\eta))$$

$$= 0 + \bar\theta a_i^T(\alpha(\eta))Z(\alpha(\eta))p_N$$

$$= 0,$$

by construction of $Z(\alpha(\eta))$ to be orthogonal to $a_i(\alpha(\eta))$, $i = 1,2,\ldots,m$. The arc $\alpha(\theta)$ is therefore a feasible arc in some neighborhood of x^*. ■

The essence of the construction of the arc is the continuity of the matrix function Z in a neighborhood of x^*.

It is usual to assume that the matrix A is of full rank in a neighborhood of the solution, which implies that the constraint qualification holds, and hence the existence of finite Lagrange multipliers at a local minimum of P1.

In the unconstrained case, it was possible to derive additional conditions for a minimum based on the second derivatives of $F(x)$. The analogous results for problem P1 require the following definition.

Second-Oraer Constraint Qualification:

The seccnd-order constraint qualification holds at $\hat{x}$ if:

(1) $\{c_i\}$ are twice continucnsly differentiable;

(2) $c_i(\hat{x}) = 0$, $i = 1,2,\ldots,m$; and

(3) for every non-zero vector p satisfying

$$A(\hat{x})^T p = 0,$$

p is the tangent of a twice-differentiable arc $\alpha(\theta)$, emanating from $\hat{x}$, along which $c_i(\alpha(0)) \equiv 0$, $i = 1,2,\ldots,m$, $0 \leq \theta \leq \varepsilon$.

Theorem 4 (Second-Order Recessary Conditions for a Local Minimum of P1)

If:

(1) F and $\{c_i\}$, $i = 1,2,\ldots,m$, are twice continuously differentiable;

(2) The first- and second-order constraint qualifications hold at x^*, then: a necessary condition for x^* to be a local minimum of P1 is that for any vector p satisfying $A(x^*)^T p = 0$,

$$p \Bigg(G - \sum_{i=1}^{m} \qquad \qquad {}_{x^*} p \geq 0,$$ where the λ_i^* are such that

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

Proof: Note that conditions (1) and (2) assure that Theorem 2 holds, and hence imply the existence of λ^* satisfying $g(x^*) = A(x^*)\lambda^*$.

Assume that there exists a non-zero vector p such that $A^T p = 0$ (othe-mise, the result follows immediately). Let $\alpha(\theta)$ be the twice differentiable arc from x^* whose existence is guaranteed by the second-order constraint qualification, where $\alpha(0) = x^*$, $\frac{d}{d\theta}\alpha(0) = p$.

Let $v = \frac{d^2}{d\theta^2}\alpha(0)$. In the following, all vector and matrix functions are evaluated at $x^* = \alpha(0)$, unless otherwise specified.

Using the chain rule,

$$(a)\ \frac{d}{d\theta} c_i(\alpha(0)) = a_i^T \frac{d}{d\theta} \alpha(0) = 0,$$

since $a_i^T p = \theta$, $1 = 1,2,\ldots,m$;

$$(b)\ \frac{d^2}{d\theta^2} c_i(\alpha(0)) = a_i^T \frac{d^2}{d\theta^2} \alpha(0) + \frac{d}{d\theta}(a_i^T) \frac{d}{d\theta} \alpha(0)$$

$$= a_i^T v + p^T G_i p = 0,$$

since the $\{c_i\}$ are identically zero along the arc $\alpha(\theta)$;

$$(c)\ \frac{d}{d\theta} F(\alpha(0)) = g^T \frac{d}{d\theta} \alpha(0) = g^T p = \lambda^{*T} A^T p = 0,$$

since $g = AA^*$ and $A^T p = 0$.

For x^* 'io be a local minimum of Pl, the condition $\frac{d}{d\theta} F(\alpha(0)) = 0$ implies that $\frac{d^2}{d\theta^2} F(\alpha(0))$ must be ≥ 0. Using (c), this condition may be written as follows:

$$\frac{d^2}{d\theta^2} F(\alpha(0)) = \frac{d}{d\theta}\left(g^T \frac{d}{d\theta} \alpha(0)\right)$$

$$= g^T \frac{d^2}{d\theta^2} \alpha(0) + p^T G p$$

$$(d)\qquad = \lambda^{*T} A^T v + p^T G p \geq 0$$

(substituting $v = \frac{d^2}{d\theta^2} \alpha(0)$ and $g = A\lambda^*$).

From (b), substitute $a_i^T v = -p^T G_i p$, $i=1,2,\ldots,m$, into (d) yielding

$$p^T\left(G - \sum_{i=1}^{m} \lambda_i^* G_i\right)\Big|_{x^*} p \geq 0 \text{ for all } p \text{ satisfying } A^T p = 0. \blacksquare \qquad (1.2.3)$$

The result (1.2.3) is equivalent to a requirement that the matrix $Z^T W Z$ be positive semi-definite at x^*, where $W = G - \sum_{i=1}^{m} \lambda_i^* G_i$.

Theorem 5: If :

(1) $\{c_i\}$, $i = 1,2,\ldots,m$, are twice continuously differentiable;

(2) $c_i(\hat{x}) = 0$, $i = 1,2,..,m$; and

(3) $A(\hat{x})$ has full rank,

then the second-order constraint qualification holds at $\hat{x}$.

Proof: We must construct, for any non-zero vector p satisfying $A^T p=0$, a twice-differentiable arc enanating from $\hat{x}$, such that p is tangent to the arc, and all constraints are identically zero along the arc in some neighborhood of $\hat{x}$.

Since $A^T p=0$, p can be written as Zp_N.

Define an arc $\alpha(Q)$ by $\alpha(0) = \hat{x}$, and $\frac{d}{d\theta}\alpha(Q) = Z(\alpha(\theta))p_N$, where $Z(\alpha(\theta))$ is a particular representation of a matrix whose columns form an orthogonal basis for the null space of $A(\alpha(\theta))$, so that

$$A(\alpha(\theta))^T Z(\alpha(\theta)) = 0.$$

Since A is continuously differentiable and of full rank at $\hat{x}$, the matrix $A(\alpha(\theta))$ is also continuously differentiable and of full rank in a suitable neighborhood of $\hat{x}$. Hence the matrix function $Z(\alpha(\theta))$ is of fixed rank and continuously differentiable in the neighborhood of $\$$, for an appropriate representation of Z (see Golub and Pereyra, 1973). Since $\frac{d}{d\theta}\alpha(\theta)=Z(\alpha(\theta))p_N$, it follows that $\frac{d^2}{d\theta^2}\alpha(\theta) = \frac{d}{d\theta}Z(\alpha(\theta))p_N$, and, by the continuity argument just given, the arc is twice continuously differentiable.

Expanding each constraint along this arc yields the following:

$$c_i(\alpha(\bar{\theta})) = c_i(\alpha(0)) + a_i{}^T Zp_N \Big|_{\alpha(0)} + \frac{\bar{\theta}^2}{2}\frac{d^2}{d\theta^2}c_i(\alpha(\eta))\Big|_{0 \le \eta \le \bar{\theta}}$$

$$= 0 + 0 + \frac{\bar{\theta}^2}{2}\frac{d}{d\theta}\left(a_i^T(\alpha(\eta))\, Z(\alpha(\eta))p_N\right)\Big|_{0 \le \eta \le \bar{\theta}}$$

Since the fnnction $a_i^T(\alpha(\theta))Z(\alpha(\theta))$ is identically zero along the arc by construction, its derivative with respect tc 8 must be zero also, and therefore, we obtain:

$$c_i(\alpha(\theta)) \equiv 0,$$

in some neighborhood of $\hat{x}$. Thus, for eny **p** satisfying $A(\hat{x})^T\mathbf{p} = 0$, we can construct the required twice-differentiable arc, and the second-order constraint qualificatioc holds at $\hat{x}$. ■

The essential point of this proof is that the linear independence-and differentiability of the columns of A allow the representation of the basis of the null space to be differentiated. Sufficient conditions for x* to be a local minimum of P1 can be similarly derived, and will be stated without proof:

Sufficient Conditions for a Local Minimum of P1:

x* is a strong local minimum of P1 if:

(1) F and $\{c_i\}$ are twice continuously differentiable;

(2) $c_i(x^*) = 0$, $i=1,2,\ldots,m$;

(3) there exists a vector λ^* such that $g(x^*) = A(x^*)\lambda^*$;

(4) for every non-zero vector p such that $A(x^*)^Tp = 0$,

$$p^T(G - \sum_{i=1}^{m} \lambda_i^* G_i)\Big|_{x^*} p > 0.$$

1.2.3 Inequality Constraints

The conditions for a minimum of P2, the inequality constrained problem,, involve a subset of the constraints.

Definition 4: A constraint $c_i(x) \geq 0$ is said to be *active* at $\hat{x}$ if $c_i(\hat{x}) = 0$, and the set of indices $\mathbf{I} = \{i \mid i\epsilon(1,2,\ldots,m)$ and $c_i(\hat{x}) = 0\}$ (or the correspcnding set of constraints) is termed the *active* or *binding* set at $\hat{x}$. A constraint $c_i(x) \geq 0$ is *inactive* at $\hat{x}$ if $c_i(\hat{x}) > 0$.

We identify the active constraints at a point because feasible perturbations are restricted for them. If $c_i(\hat{x}) > 0$, there is a neighborhood of $\hat{x}$ within which c_i will remain strictly positive. Hence, only the active constraints are significant in specifying the conditions for a minimum of P2. The results for a minimum of P2 are analogous to those already given for P1, but involve only the subset of constraints active at a particular point,

Throughout this section, it will be assumed that m $(\leq \ell)$ of the inequality constraints of P2 are active at any point, and the indices of the active constraints are given by the index set I. The matrix $\hat{A}(x)$ will denote the n by m matrix whose columns are the gradients of the constraints active at x.

Theorem 6: (Existence of Lagrange Multipliers for inequalities)

If:

(1) F and $\{c_i\}$, $i=1,2,\ldots,\ell$, are once continuously differentiable;

(2) $c_i(x^*) \geq 0$, $i=1,2,\ldots,\ell$;

(3) for any non-zero vector p satisfying $\hat{A}(x^*)^T p \geq 0$,

it also holds that $g^T(x^*)p \geq 0$, then there exists a vector A, of length m, sach that $g(x^*) = \hat{A}(x^*)\lambda$, and $\lambda_i \geq 0$; thus, $g(x^*)$ is a non-negative linear combination of the gradients of the active constraints.

Proof: The proof is immediate from application of Farkas' Lemma to the set of vectcrs $\{a_i\}$, $i \in I$, and the vector g. This result differs from the analogous result for P1 because the multipliers corresponding to inequality constraints are required to be non-negative. ∎

It is possible to include $\{a_i\}$ for the inactive constraints in the expression for g by assigning a zerc Lagrange multiplier to an inactive constraint; however, the convention generally followed is that a Lagrange multiplier corresponds only to an active ccnstraint.

The conditions under which the first- and second-order constraint qualifications hold at a point $\hat{x}$ are defined in teras of the set of active constraints at $\hat{x}$. The definitions are similar to those for the equality problem, with the following differences: for the first-order constraint qualification to hold, any non-zero p satisfying $\hat{A}^T p \geq 0$ must be tangent to a feasible arc; and for the second-order constraint qualification, any non-zero vector p satisfying $\hat{A}^T p = 0$ must be the tangent of an arc along which all constraints active at $\hat{x}$ remain identically zero. The linear independence of the columns of $\hat{A}(\hat{x})$ implies that both the first- and second-order constaint qualifications hold at $\hat{x}$.

The first- and second-order necessary conditions for x^* to be a local minimum of P2 are the same as for P1, where the set of active constraints is treated as a set of equality constraints, with the significant restriction that all components of the vector λ^* of multipliers satisfying $g(x^*) = A(x^*)\lambda^*$ must be non-negative.

The <u>sufficient</u> conditions that x^* is a strict iocal minimum of P2 are the same as those for P1, where the set of active constraints at x^* is treated as a set of equality constraints, with the following additional conditions:

(a) $\lambda_i^* \geq 0$, $i=1,2,\ldots,m$; and

(b) for any p satisfying $\hat{a}_i^T p = 0$, $\lambda_i^* > 0$, and $\hat{a}_i^T p \geq 0$, $\lambda_i^* = 0$, then $p^T(G - \sum_{i \in I} \lambda_i^* G_i)p > 0$, where the summation is over the active constraints only. The extra condition on **p** arises because of the possibility of a zero Lagrange multiplier for an active constraint.

1.3 Model Computational Algorithm

The class of algorithms to be considered throughout for iteratively locating a minimum of P1 or P2 has the following structure:

```
(Check whether x, the current estimate of the solution, satisfies
 the termination criteria)
while x does not satisfy these criteria repeat
⟨ 1. Compute a search direction, p, to satisfy some specified
     conditions;
  2. Compute a step $\alpha$ along p such that $x + \alpha p$ satisfies some
     given criteria;
  3. $x \leftarrow x + \alpha p$;
⟩
```

The algorithms to be described differ in the criteria specified for each computation, and in the methods used to determine the search direction and step length.

1.4 Unconstrained Minimization

The methods to be considered are designed to determine a local unconstrained minimum of the function F(x).

1.4.1 Descent Condition

For the unconstrained problem, a natural measure of the progress made by an iteration is provided by the decrease in F during the iteration. The criterion of the value of F allows a consistent decision as to which point of a set is the "best" estimate of the solution, x^*. All algorithms tc be described require a decrease in the value of F from iteration to iteration, and are said to satisfy the "descent condition".

1.4.2 Choice of Search Direction

The Taylor series expansicn of $F(x)$ about any point provides insight into the choice of search direction:

$$F(x+\alpha p) = F(x)+\alpha g(x)^T p + \frac{\alpha^2}{2} p^T G(x)p+O(|\alpha|^3||p||^3).$$

If $g(x)\neq 0$, $F(x+\alpha p)$ will be less than $F(x)$ for all sufficiently small $\alpha > 0$ only if $g(x)Tp < 0$. A direction p satisfying $g(x)Tp < 0$ is said to be a descent direction at x for $F(x)$. All algorithm to be considered require that the search direction be a descent direction.

Method of Steepest Descent (see Ostrowski, 1966)

The classical method of steepest descent chooses the negative gradient of F as the search direction, so that $p = -g$. This choice minimizes the first term in the Taylor series expansicn of $F(x+\alpha p)$ for fixed cy and $||p||_2$, i.e., $p = -g$ solves the problem:

minimize pTg

subject to $p^T p = \beta^2$, for β a fixed scalar.

The method of steepest descent is a poor choice for general functions. Its asymptotic rate of convergence is linear, and it may converge arbitrarily slowly even for quadratic functions.

Newton's Method (see Ortega and Rheinboldt, 1970)

The general quadratic function:

$$q(x) = a + b^T x + \frac{1}{2}x^T Qx,$$

hss a unique minimum if its Hessian matrix, Q, is positive definite. The minimum occurs at x^* where the gradient vector is zero, i.e.,

x^* satisfies:

$$b + Qx^* = 0$$

This relationship for x^* implies that from any starting point, say $\hat{x}$, the minimum of a suitable quadratic function can be determined by a unit step along the search direction **p** that solves the linear system:

$$Qp = -b-Q\hat{x}.$$

The expression on the right-hand side is the negative gradient of the quadratic function at $\hat{x}$.

Any sufficiently smooth function should be well approximated in some neighborhood by a truncated Taylor series. If the Hessian matrix at the current point is positive definite, the step p to the minimum of the local quadratic approximation will be given by the solution of:

$$Gp = -g, \tag{1.4.1}$$

where g is the current gradient and G is tne Hessian matrix. The choice cf search direction to satisfy (1.4.1) is usually called "Newton's method". For certain classes of functions, Newton's method displays a quadratic rate of convergence close to the solution, and is highly successful.

However, for a number of reasons, Newton's method as defined by (1.4.1) is inadequate for many functions. To ensure a decrease in $F(x)$, some sort of linear search along p is required, since the local quadratic approximation may be poor far from the minimum. If the current Hessian matrix, G, is singular, the search direction is not defined uniquely. If G is indefinite, the local quadratic approximaticn does not have a unique minimum, and the best definition of search direction in this case is not clear.

Quasi-Newton Methods (See Broyden, 1972)

The algorithms known as "quasi-Newton" methods were originally derived from the properties of quadratic functions (see Davidon, 1959; Fletcher and Powell, 1963). Quasi-Newton methods do not calculate the Hessian matrix at each point, but rather maintain an approximation to the Hessian, which is altered at each iteration based on the variation in the gradient as the iterations proceed. The search direction is then the solution of the linear system $Bp = -g$, where B is the current Hessian approximation.

There are a wide variety of these methods, with different procedures for updating the Hessian approximation. With exact arithmetic, if the step α chosen at each iteration is the step to the minimum of $F(x)$ along p, most quasi-Newton methods can be shown to converge to the minimm of a positive definite quadratic function in a finite number of steps (a property sometimes confusingly called "quadratic termination").

Methods that may alter the search direction during an iteration

A further method for unconstrained minimization, based on an idea proposed originally by Levenberg (1944) in the context of nonlinear Least squares, assumes that a unit step will always be taken along the computed search direction. The search direction for such algorithms is given by the solution of the linear system:

$$(G + \gamma I)p = -g, \qquad (1.4.2)$$

where the parameter γ, $\gamma > 0$, may be adjusted during an iteration, and from iteration to iteration as well. As $\gamma \rightarrow 0$, the search direction from (1.4.2) approaches the Newton step; for large γ, the search direction approaches a small multiple of the negative gradient. The adjustment of γ during an iteration is based on a decision as to whether a unit step along the current direction results in a sufficient decrease in $F(x)$; γ is altered from iteration to iteration based on the success of the previous choice of γ, and γ should ideally approach zero as the iterates approach the solution.

Another method that determines tne search direction as a compromise between a Newton (or quasi-Newton) step and a multiple of the negative gradient has been proposed by Powell (1970), and recently modified by Dennis and Mei (1975). As in the Levenberg-type algorithms, several search directions may be computed during an iteration, based on whether a unit step yields a sufficient decrease in $F(x)$.

1.4.3 Choice of Step Length

In some algorithm for unconstrained minimization, a unit step is always taken along the search direction (see Section 1.4.2), so that only the procedure for calculating the search direction is significant. For other methods, however, such as Newton-type or quasi-Newton, the two choices are independently carried out: a descent search direction is obtained, and the step to be taken along it is determined to satisfy various criteria. The general procedure of computing α such that, for a specified direction p, $F(x+\alpha p)$ satisfies certain conditions with respect to $F(x)$, is called a linear search, since it is a one-dimensional problem.

Most algorithms for unconstrained optimization include some criteria to assure that the step length chosen will guarantee "satisfactory" progress toward the solution. Especially in theoretical presentations of algorithms, It Is often stated that $F(x+\alpha p)$ snould be a minimum with respect to α; this requirement leads to a "perfect" or "exact" line search. In practical implementations of the algorithms, however, the standards for deciding on an acceptabie value of α are generally less stringent. The criteria for terminating a linear search will not be discussed further here (see Chapter 3 for additional comments); usually the step length can be considered an approximation to the step to the minimum of $F(x)$ along p.

1.4.4 Summary

The methods described for unconstrained optimization have all corresponded to the pattern of the model algorithm. The decrease in $F(x)$ from iteration to iteration is consistently used as a measure of a method's progress toward the solution. The most successful algorithms are based on approximating $F(x)$ by a positive definite quadratic function, since this approximation should be accurate near the minimum.

1.5 Linear Equality Constraints

The problem to be solved is:

$$\text{minimize } F(x)$$
$$\text{subject to } A^T x = b,$$

where A^T is m by n, $m \leq n$.

The necessary conditions derived in Section 1.2 for a local minimum of P1 specify that, at x^*:

1) $A^T x^* = b$;

2) there exists a vector λ^* such that $g(x^*) = A\lambda^*$;

3) $p^T G p \geq 0$ for all p such that $A^T p = 0$.

It should be noted that the constraint qualifications always hold for linear constraints. Given a matrix Z, whose columns form an orthogonal basis for the null space of A, conditions 2) and 3) can be rewritten as: ,

2') $Z^T g(x^*) = 0$;

3') $Z^T GZ$ is positive semi-definite.

If the matrix A has Pull rank, it is always possible to find an initial feasible point, y, where all constraints are satisfied (so that $A^T y = b$). If any move from y is made along directions that lie in the null space of' the columns of A, then all subsequent points will also satisfy the constraints.

Any vector in the null space of A can be written as Zv, for Z as described above. All vectors x that satisfy $A^T x = b$ can, therefore, be written as $x = y + Zv$, where y satisfies $A^T y = b$, an:! the constrained problem can be solved by determining the vector v^* of length (n-m) such that $F(y + Zv^*)$ is a minimm with respect to v. The linearly constrained problem can thus be considered as an unconstrained problem in a reduced number of variables.

1.5.1 Descent Condition

Because the initial constrained pro?;lem has been transformed into an unconstrained problem of minimizing F(x) in a subspace, the descent condition with respect to F can still be required for any iteration after an initial feasible point is found. This requirement is possible because the constraints are automatically satisfied at every iteration, and only the decrease in F is significant in measuring improvement of the iterates.

1.5.2 Choice of Search Direction

As in the unconstrained case, methods for choosing the search direction are based on the properties holding at $F(x^*)$. A fundamental complication is that the Taylor series expansion of F cannot be used in a purely straightforward way. The feasibility of any perturbation must be considered, and the Hessian matrix of F need not be positive definite in a neighborhood of the minimum, nor even at the minimm itself.

However, with linear equality constraints, from an .initial feasible point it is possible to remain feasible simply by moving only along appropriately chosen directions. Hence, the behavior of F(x) along such directions, any of which may be written as $p = Zv$, can be examined:

$$F(x+Zv) = F(x) + v^TZ^Tg + \frac{1}{2} v^TZ^TGZv + \dots$$

At x^*, the "projected gradient," Z^Tg, must be zero, and Z^TGZ must be positive semi-definite. If F(x) is a quadratic with Hessian G, and Z^TGZ is strictly positive definite, the vector v such that Zv is the step to the minimum of $F(x + Zv)$ is given by the solution of:

$$Z^TGZv = -Z^Tg. \qquad (1.5.1)$$

A general algorithm, where the search direction is given by Zv, and v satisfies (1.5.1) for the current gradient and Hessian, is a Newton-type method analogous to the unconstrained case. If Z^TGZ is positive definite, then $g^Tp = -g^TZ(Z^TGZ)^{-1}Z^Tg$, and p is a descent direction for F.

There are many other methods for solving linearly constrained problems (see Gill and Murray, 1974b), and details will not be given here. The important point is that, because the constrained problem can be transformed into an equivalent unconstrained problem, methods based on unconstrained techniques can be directly applied, by considering the behavior of the o'rjective function along a limited set of directions. For this problem, it is always possible to determine a search direction such that a suitable step simultaneously maintains feasibility and decreases the objection fnnction.

1.5.3 Choice of Step Length

Because a search direction can be computed along which unlimited movement will not violate the constraints, the criteria for choosing the step length can be based only on $F(x)$, and remain the same as in whatever unconstrained method is used to choose the search direction.

1.6 Linear Inequality Constraints

The problem to be considered is:

$$\text{minimize } F(x)$$

$$\text{subject to } A^T x \geq b,$$

where A^T has ℓ rows.

The conditions that must be satisfied at x^*, a local minimum, involve the columns of A correspondicg to the constraints active at x^*. Let $\hat{A}$ denote the matrix whcse columns correspond to the active constraints, and let $\hat{Z}$ be a matrix whose columns form an orthogonal basis for the null space of $\hat{A}$. At x^*, the following must hold:

1) $A^T x^* \geq b$ (all constraints satisfied);

2) $g(x^*) = \hat{A}\lambda^*$, with all $\lambda^*_i \geq 0$;

3) $\hat{Z}^T G(x^*)\hat{Z}$ must be positive semi-definite.

The linear inequality constrained problem is more complicated than the linear equality constrained problem because: (1) it is not possible, in general, to determine immediately a point satisfying all the constraints; and (2) the set of constraints active at x^* is unknown. The problem (1) of finding an initial feasible point will not be discussed here in any detail; in the case of linear programming (solving the problem above for a linear objective function), this sub-problem is known as "Phase I". It can be solved by minimizing a function that represents the departure of a point from feasibility, e.g., the sum of the magnitudes of the constraint violations (see Gill and Murray, 1974b). Since methods are available for finding a feasible point if one exists, it will be assumed henceforth that an initial feasible point is given. The question posed by (2) of finding the set of active constraints will be answered as the solution is determined.

1.6.1 Choice of Search Direction

Most algorithms for computing the search direction for this problem are based on an "active set" strategy. With such a strategy, a subset of the constraints - the "active set" - is treated as a temporary set of equality constraints. Under appropriate conditions, the search direction can then be chosen to decrease the objective function and to remain feasible with respect to the currently active constraints. Let $\hat{A}$ denote the full-rank matrix whose columns correspond to those constraints currently considered active, with the usual significance for $\hat{Z}$. Any direction which is orthognnal to all currently active constraints can be written' as $p = \hat{Z}v$ for some v. Expanding F(x) along such a direction gives:

$$F(x+\hat{Z}v) = F(x) + g^T\hat{Z}v + \frac{1}{2} v^T\hat{Z}^TG\hat{Z}v + \ldots$$

If g is not in the range of $\hat{A}$, there exists a direction $\bar{p} = \hat{Z}\bar{v}$ such that $\hat{A}^T\bar{p} = 0$ and $\bar{p}$ is a descent direction for F, i.e., $\bar{p}^T g < 0$. Otherwise, g is in the range of $\hat{A}$, so that $\hat{Z}^T g = 0$, and $g = \hat{A}\hat{\lambda}$ for some $\hat{\lambda}$. If any component of $\hat{\lambda}$ is negative, say $\hat{\lambda}_i < 0$, a direction $\hat{p}$ exists such that $a_i^T \hat{p} > 0$, $g^T\hat{p} < 0$, and $a_j^T \hat{p} = 0$ for all other active constraints. A move along $\hat{p}$ essentially removes the ith constraint from the active set, and $\hat{p}$ is a feasible descent direction with a reduced active set. Except at x*, it is always possible to find a feasible descent direction, although this determination may involve altering the current active set of constraints.

Except when the active set is modified, the calculation of search direction for a linear inequality constrained problem is essentially the same procedure as for an equality constrained problem, where the current active set specifies the equalities. Thus, the search direction is chosen by application of some suitable algorithm for finding the unconstrained minimum of F in a subspace.

1.6.2 Choice of Step Length

The step to be taken along a search direction depends on two considerations: (1) the criteria used to choose the step based on the decrease in F; and (2) whether the next point remains feasible with respect to constraints not previously active. The new question posed by (2) can be answered fairly easily; with linear constraints, it is possible to compute exactly where each constraint will become zero along a given direction. At each iteration, the search direction produced is a descent direction for F, and only positive steps will be taken along it. If $a_i^T p \geq 0$, where the ith constraint is $a_i^T x - b_i \geq 0$, the constraint is nondecreasing along p, and, if currently inactive, will not reach zero for any positive step. If

$a_i^T p < 0$, the step γ_i to the constraint's zero along p, such that

$$a_i^T (x+\gamma_i p) = b_i,$$

is given by:

$$\gamma_i = \frac{(b_i - a_i^T x)}{a_i^T p} .$$

The values γ_i can be computed for all currently inactive constraints decreasing along p, and $\gamma^* = \min\{\gamma_i\}$ is then the maximum positive step'that will not violate any constraint.

The step length algorithm for a linear inequality constrained problem must determine whether an acceptable step with respect to the objective function (satisfying the criteria in the unconstrained context) is achieved at $\alpha < y^*$. If so, no new constraints become active at the next point; if not, the step y* will be taken, and a previously inactive constraint is added to the active set.

1.6.3 Summary

The minimization problem with linear inequality constraints can be solved by using the methods given for the equality case. If the constraints are not inconsistent, an initial feasible point can be found. Given a set of currently active constraints to be treated as equalities, it is possible to determine feasible descent directions, perhaps simultaneouslv altering the specification of the active set, with a decrease in F(x) at every iteration. The step along any search direction at which an inactive constraint becomes zero can be obtained directly, so that there is no difficulty with maintaining feasibility. Because the search direction at each iteration generates only feasible points with respect to the current active set, the linear inequality constrained problem can be transformed into a sequence of unconstrained problems, each of which has a transparent relationship to the original problem.

1.7 Nonlinear Constraints - New Considerations

For consistency, it will be assumed henceforth that all the constraints of a problem are nonlinear. If even one constraint function is nonlinear, the problem is considered to be nonlinearly constrained; but since advantage can be taken of linearity in the constraint functions, linear constraints are assumed to be treated separately.

If an objective function is not quadratic, the unconstrained minimization problem cannot, in general, be solved without iteration. A corresponding increase in complexity occurs with nonlinear constraint functions. Any nonlinearity in a constraint function means that, at a point where the constraint is zero, it is nc longer possible, in general, to find a straight line along which the cmstraint will remain zero for even an arbitrarily small move. Hence, a move along a search direction that tends to decrease F(x) may inevitably violate some of the currently active constraints.

Algorithms for the nonlinearly constrained problem must consider not only how to decrease F(x), but also how to attain or maintain feasibility. The following new questions must be answered in designing algorithms for the nonlinearly constrained problem:

(1) What criteria should be used to determine when an iteration has been "successful"? The previous classes of algorithms were able to use the consistent criterion of the value of F(x) to decide which of two points is a "better" estimate of the solution., In the nonlinearly constrained case, the criteria of success must include both F(x) and the deviation from feasi-

bility, and it is not obvious how to balance these two measures.

(2) On what basis should the search direction and step length be chosen? Should p be a descent direction for $F(x)$? Minimize a local quadratic approximation to $F(x)$, either in the full space or along a limited set of directions? Maintain feasibility as closely as possible? Decrease infeasibility? Should the step length α be chosen to approximate the minimum of some function along p? If so, which function?

A possible approach to answering these questions is suggested by an idea that emerged from the linearly constrained cases - transform the constrained problem into a sequence of less complex sub-problems (or possibly a single sub-problem) whose ultimate solution is the solution of the constrained problem. If a su5-problem can be devised so that, at Its solution, satisfactory progress has been made toward the constrained solution according to some criteria, then the well-defined framework of the sub-problem can be adopted until it is solved.

The remaining chapters will discuss some of' the numerous methods that have been proposed for solving the nonlinearly constrained problem, with particular attention to the ways in which the sub-problems are constructed, and the corresponding procedures for measuring progress, constructing search directions, and determining the step length.

It will be seen that the general minimization problem with nonlinear constraints contains complications thet do cot exist with linear constraints. Consequently, it is our view that a fundamental departure is advisable from the strategy of algorithms devised for the linearly constrained problems.

CHAPTER 2. Penalty and Barrier Function Methods

CHAPTER 2

PENALTY AND BARRIER FUNCTION METHODS

2.1 Penalty Functions - Introduction

The problem to be considered first is the nonlinear equality constrained problem:

P1: minimize $F(x)$

subject to $c_i(x) = 0$, $i=1,2,...,m$.

An approach to solving P1, originally proposed by Courant (1943), is to construct a sequence of related unconstrained problems whose solution in the limit is the solution of P1. With such a transformation of P1, the equality constrained problem could be solved simply by repeated application of an unconstrained algorithm to a set of modified objective functions.

In general, a local unconstrained minimum of F(x) will occur at a point not satisfying the constraints, or, alternatively, F(x) may decrease without bound. The solutions of the proposed sequence of unconstrained sub-problems will converge to x* only if the modified objective functions include a term that enforces feasibility in the limit. One possible choice for such a term is a function that assigns a positive measure to the constraint violation (a "penalty" for infeasibility), If the term measuring infeasibility is assigned an increasingly large weight in each successive unconstrained sub-problem, it seems plausible that the limit of the solutions will satisfy all constraints.

2.2 Penalty Functions - Theory

2.2.1 Definition

The term "penalty function'' can be defined with wide generality (see Fiacco and McCormick, 1968 ; Ryan, 1974), but throughout the present discussion only the following penalty function will be considered :

$$P(x,\rho) = F(x) + \frac{\rho}{2} c(x)^T c(x).$$

The term cTc is clearly a measure of infeasibility, and is sometimes called the "quadratic loss function" (the multiple 1/2 is included to avoid the appearance of a factor of 2 in the derivatives). The positive parameter ρ will be termed the "penalty parameter", and is increased to infinity in the limit to enforce feasibility. This parameter is often written as $\frac{1}{r}$, where $r \rightarrow 0$, and r is defined as the "penalty parameter"; but it seems more consistent for the parameter that enforces the increasing penalty to be increasing. Computationally, the treatment is identical. Since neither $+\infty$ nor 1/0 exists in any computer's number system, the limit is never taken, and ρ is made large enough (or r small enough) to satisfy the specified convergence criteria. It is possibie to assign a different penalty parameter to each constraint, thereby varying the influence of each violation; however, only a single parameter will be considered here.

2.2.2 Description of Algorithm

The penalty function method for finding x*, a local constrained minimum of P1, is the following:

```
(Choose a starting point, called $x^*(\rho^{(0)})$ and an
 initial value of $\rho$, $\rho^{(1)}$ ;
 $i \leftarrow 1$; 〉
while (the conditions for a constrained minimum are
  not satisfied by $x^*(\rho^{(i-1)})$)
repeat 〈
```

```
    Find $x^*(\rho^{(i)})$, tbe unconstrained minimum of $P(x,\rho^{(i)})$,
    starting at $x^*(\rho^{(i-1)})$;
    Choose $\rho^{(i+1)} > \rho^{(i,}$;
    $i$ t $i + 1$;
    ⟩
```

The use of such a penalty function algorithm provides straightforward answers to some of the questions posed in Section 1.7 of algorithm design for nonlinearly constrained problems. If the constrained minimum has not been determined with sufficient accuracy, a penalty function method involves the unconstrained minimization of a particular function, and at each iteration the computation (choice of search direction and step length) is directed toward solution of this sub-problem. However, a new decision - how to choose and vary the penalty parameter - has been introduced.

Figure 2.1 illustrates the convergence of the sequence $x^*(\rho)$ to a constrained solution.

2.2.3 Properties of Penalty Function Methods

Convergence

Fiacco and McCormick (1968) present a proof that under reasonably general circumstances, the sequence $x^*(\rho)$, the minima of successive penalty functions, converges to x^* as $\rho \to \infty$. Penalty function methods will converge even when the first-order constraint qualification does not hold at the limit point, and hence can determine minima that do not satisfy the Kuhn-Tucker conditions.

The convergence proof is quite general, and requires few conditions on the objective and constraint functions. However, the proof shows only the existence of a compact set including x^* within

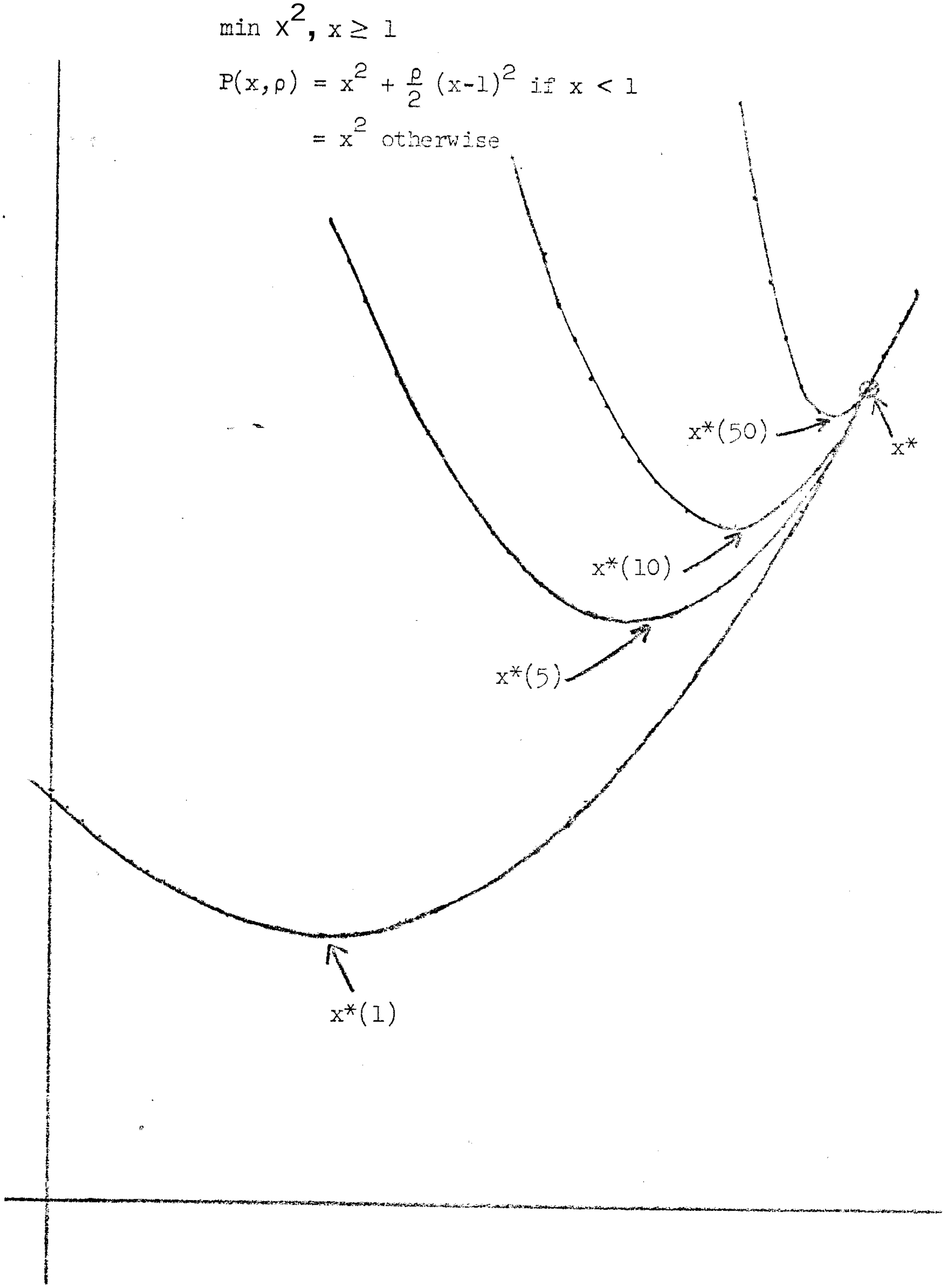

min x^2, $x \geq 1$
$P(x,\rho) = x^2 + \frac{\rho}{2}(x-1)^2$ if $x < 1$
$= x^2$ otherwise
x*(50)
x*
x*(10)
x*(5)
x*(1)

FIGURE 2.1

which the sequence $x^*(\rho)$ will converge to x^* as $\rho \rightarrow \infty$; in other words, $x^*(\rho)$ is the minimum of $P(x,\rho)$ restricted to a finite domain.

The practical implication of this. restriction is that it cannot be assumed that a suitable unconstrained minimum of $P(x,\rho)$ will be found for any starting point. The effect of applying a penalty function transformation to a constrained problem is to create a set of <u>local minima</u>. For ρ sufficiently large, and within a bounded region including x^*, the sequence of particular local minima of $P(x,\rho)$ will converge to x^*. However, the unconstrained algorithm used to minimize $P(x,\rho)$ may fail to converge to the desired local minimum if the starting point is not within the bounded region.

This limitation is illustrated in an example given by Powell (1972):

minimize x^3

subject to $x-1 = 0$.

The penalty function, $P(x,\rho) = x^3 + \frac{\rho}{2}(x-1)^2$, is displayed in Figure 2.2. The solution to the problem is clearly $x^* = 1$, and the penalty function has a local minimum at

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

However, if the starting point is not close enough to x^*, an unconstrained algorit'hm will fail to converge to x^*, since for unrestricted x,$P(x,\rho)$ is unbounded below for any ρ.

<u>Properties of successive $x^*(\rho)$</u>

The minima of successive $P(x,\rho^{(k)})$ have the following properties, where $x^{(k)}$ denotes $x^*(\rho^{(k)})$, $F^{(k)}$ denotes

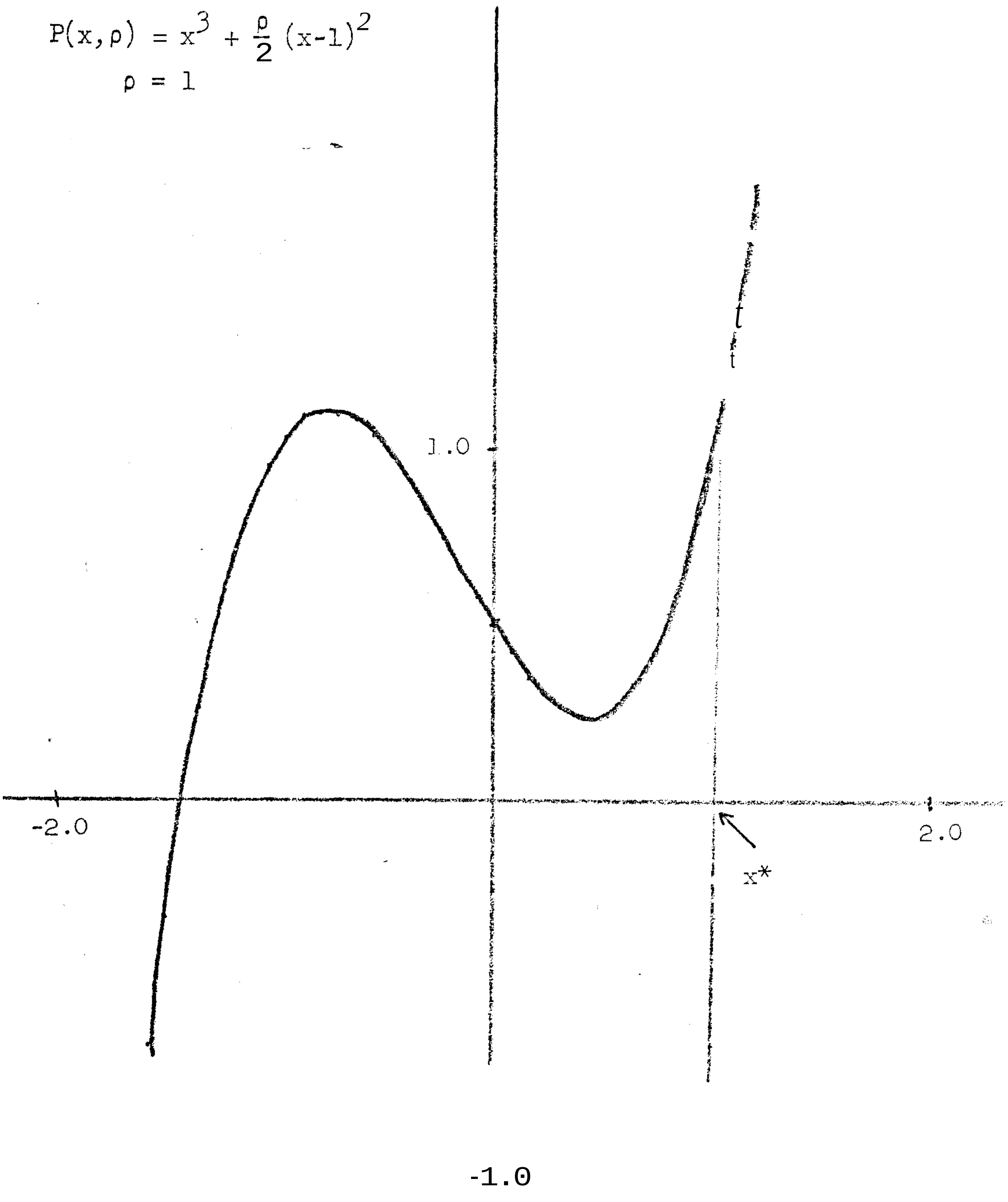


FIGURE 2.2

$F(x^{(k)})$, and $c^{(k)}$ denotes $c(x^{(k)})$.

1. $P(x^{(k)}, \rho^{(k)})$, $k = 1,2,\ldots,$ is nondecreasing.

Proof: By definition

$$P(x^{(k-1)},\rho^{(k-1)}) \le P(x^{(k)},\rho^{(k-1)});$$

hence

$$F^{(k-1)} + \frac{\rho^{(k-1)}}{2} c^{(k-1)^T} c^{(k-1)} \le F^{(k)} + \frac{\rho^{(k-1)}}{2} c^{(k)T} c^{(k)}$$

$$< F^{(k)} + \frac{\rho^{(k)}}{2} c^{(k)T} c^{(k)}$$

(if $c^{(k)T} c^{(k)} \ne 0$) since $\rho^{(k)} > \rho^{(k-1)}$

2. $F(x^{(k)})$, $k=1,2,\ldots,$ is nondecreasing.

Proof: It follows from the definition of $x^{(k-1)}$ and $x^{(k)}$ that:

$$F(k-1) + \frac{\rho^{(k-1)}}{2} c^{(k-1)T} c^{(k-1)} \le F(k) + \frac{\rho^{(k-1)}}{2} c^{(k)T} c^{(k)}$$

and

$$F^{(k)} + \frac{\rho^{(k)}}{2} c^{(k)T} c^{(k)} \le F^{(k-1)} + \frac{\rho^{(k)}}{2} c^{(k-1)T} c^{(k-1)}$$

Multiplying the first inequality by $\frac{\rho(k)}{\rho(k-1)}$, and adding it to the second, gives, after re-arranging:

$$\left(\frac{\rho^{(k)}}{\rho^{(k-1)}} - 1\right) F^{(k-1)} \le \left(\frac{\rho^{(k)}}{\rho^{(k-1)}} - 1\right) F^{(k)},$$

which implies that $F^{(k-1)} \le F^{(k)}$, since $\frac{\rho^{(k)}}{\rho^{(k-1)}} > 1$.

3. $c^{(k)T} c^{(k)}$, k = 1,2, ..., is nonincreasing.

Proof: $F^{(k)} + \frac{\rho^{(k)}}{2} c^{(k)T} c^{(k)} \le F^{(k-1)} + \frac{\rho^{(k)}}{2} c^{(k-1)T} c^{(k-1)}$,

so that

$$F^{(k)} - F^{(k-1)} + \frac{\rho^{(k)}}{2} c^{(k)T} c^{(k)} \le \frac{\rho^{(k)}}{2} c^{(k-1)T} c^{(k-1)};$$

and since, from property 2, $F^{(k)} - F^{(k-1)} \ge 0$,

$$c^{(k)T} c^{(k)} \le c^{(k-1)T} c^{(k-1)}.$$

These properties show that the successive $x^*(\rho)$ display a decreasing measure of infeasibility and an increase in $F(x)$ as the iterates approach the feasible solution x^*.

The minima of successive penalty functions provide a set of points from which no move exists tnat simultaneously decreases F and c^Tc. At $x^*(\rho)$, the minimum of $P(x,\rho)$:

$$\nabla P(x,\rho) = g + \rho Ac = 0.$$

Any direction p along which cTc is decreasing from $x^*(\rho)$ satisfies $p^TAc < 0$; but since $Ac = \frac{-1}{\rho} g$, it follows that $p^Tg > 0$, and any such p will locally increase $F(x)$.

Lagrange Multiplier Estimates

The condition:

$$g = -\rho Ac,$$

which holds at $x^*(\rho)$, is of the same form as the necessary condition for a minimum at x^* when $A(x^*)$ has full rank, i.e.,

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

The quantities $\lambda_i(\rho) = -\rho c_i(x^*(\rho))$ are thus analogous to the Lagrange multipliers at x^*, since they are the coefficients in the expansion of g as a linear combination of the columns of A. It can be shown that under suitable conditiocs - for example, when the sufficiency conditions given in Section 1.2 hold at x^*, and $A(x^*)$ has full renk - then $\lim_{\rho \to \infty} \lambda_i(\rho) = \lambda_i^*$.

Trajectory of $x^*(\rho)$

Under suitable assumptions, the penalty parameter can be regarded as an independent variable defining a trajectory of values $x^*(\rho)$, with corresponding multiplier functions $\lambda(\rho)$, converging respectively to x^*-and λ^*. The following result is proved in Fiacco and McCormick (1968). If:

(1) F and $\{c_i\}$ are twice differentiable;

(2) $A(x^*)$ has full rank;

(3) The sufficient conditions given in Section 1.2 for a constrained minimum are satisfied at x^*;

Then for ρ sufficiently large there exist continuously differentiable functions $x^*(\rho)$, $\lambda(\rho)$, such that $\lim_{\rho \to \infty} x^*(\rho) = x^*$, $\lim_{\rho \to \infty} \lambda(\rho) = \lambda^*$, and for any finite ρ,$x^*(\rho)$ is a local minimm of $P(x,\rho)$.

When the trajectory $x^*(\rho)$ exists, it is possible to use extrapolation techniques to predict x^* based on previous values of $x^*(\rho)$. This technique will not be discussed further here; for details, see Fiacco and McCormick (1968).

The trajectory defined by $x^*(\rho)$ has several interesting properties\. In order to discuss the conditions at the limit, we introduce the variable $r = \frac{1}{\rho}$, use the notation $x^*(\rho) \equiv x^*(r)$, and take the limit as $r \to 0$. It is possible to express $x^*(r)$ in terms of its expansion about x^* as:

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

wnere $y = \lim\limits_{r \to 0} \dfrac{x^*(r)-x^*}{r} = \lim\limits_{r \to 0} \dfrac{dx^*(r)}{dr}$

At $x^*(r)$, $r\nabla P(x,r) = rg + Ac \equiv 0$.

Differentiating this identity with respect to r at $x^*(r)$ gives:

$$\frac{d}{dr}(rg + Ac) =$$

$$g + \frac{d}{dx^*}(rg + Ac)\frac{dx^*(r)}{dr} =$$

$$g + (rG + \sum_{i=1}^{m} c_i G_i + AA^T)\frac{dx^*(r)}{dr} = 0.$$

As $r \to 0$, by definition $c_i \to 0$, and the above becomes:

$$g(x^*) + A(x^*)A(x^*)^T y = 0,$$

where $y = \lim\limits_{r \to 0} \dfrac{dx^*(r)}{dr}$. If $A(x^*)$ has full rank, $g(x^*) = A(x^*)\lambda^*$, so that by substituting for $g(x^*)$ and cancelling $A(x^*)$, the result is:

$$A(x^*)Ty = -\lambda^*. \qquad (2.2.1)$$

This relationship can also be obtained by the following alternative derivation. At $x^*(r)$, $\lambda(r) = \frac{-1}{r}c(x^*(r))$. Expanding the ith constraint and multiplier estimate about x^*, and letting the vector u_i denote $\frac{\partial \lambda_i}{\partial x}$, we obtain:

$$\lambda_i(x^*(r)) = \frac{-1}{r}c_i(x^*(r))$$

$$\lambda_i(x^*) + r\,u_i(x^*)^T y = \frac{-1}{r}c_i(x^*) - a_i(x^*)^T y + O(r^2)$$

$$= -a_i(x^*)^T y + O(r^2),$$

since $c_i(x^*) = 0$ for all i. Taking the limit as $r \to 0$ gives, as before:

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

If $\lambda_i^* \neq 0$, it follows that $a_i^T y \neq 0$. The penalty function trajectory's approach to x^*, therefore, does not lie in a tangent plane for any constraint with a non-zero Lagrange multiplier. For sufficiently small r, a similar result will hold for the step from $x^*(r)$ to $x^*(\hat{r})$, $\hat{r} < r$, by the continuous differentiability of $x^*(r)$. This characterization of the path of iterates to x^* contrasts to that of other algorithms, for example, "projected gradient" methods, which generate a tangential approach to the solution. Figure 2.3 displays the typical approach of e penalty function trajectory to x^*.

The non-zero scalar product between a_i and y implies that the linear term will dominate the local Taylor series expansion of c_i along the trajectory. If $c_i(x+\alpha p)$ is approximated by $c_i+\alpha a_i^T p$, and $\frac{|a_i^T p|}{\|a_i\|\,\|p\|}$ is very small, the linear approximation is poor: and higher-order information is required to model accurately the behavior of c_i along p. In Chapter 5, there is further discussion of the significance of the non-tangential approach to x^* of the penalty function trajectory, including consideration of how this property can be exploited in algorithm design.

To derive an expression for y, we use the identity for $\nabla P(x^*(\rho),\rho)$:

$$\begin{aligned} 0 &= \nabla P(x^*(\rho),\rho) \\ &= g(x^*(\rho)) - A(x^*(\rho))\,\lambda(x^*(\rho)), \end{aligned}$$

and expand each function about x^*, using $x^*(\rho) = x^* + \frac{1}{\rho} y + O(\frac{1}{\rho^2})$. In the following, all matrix and vector functions are evaluated at x^* unless otherwise specified. The above identity becomes:

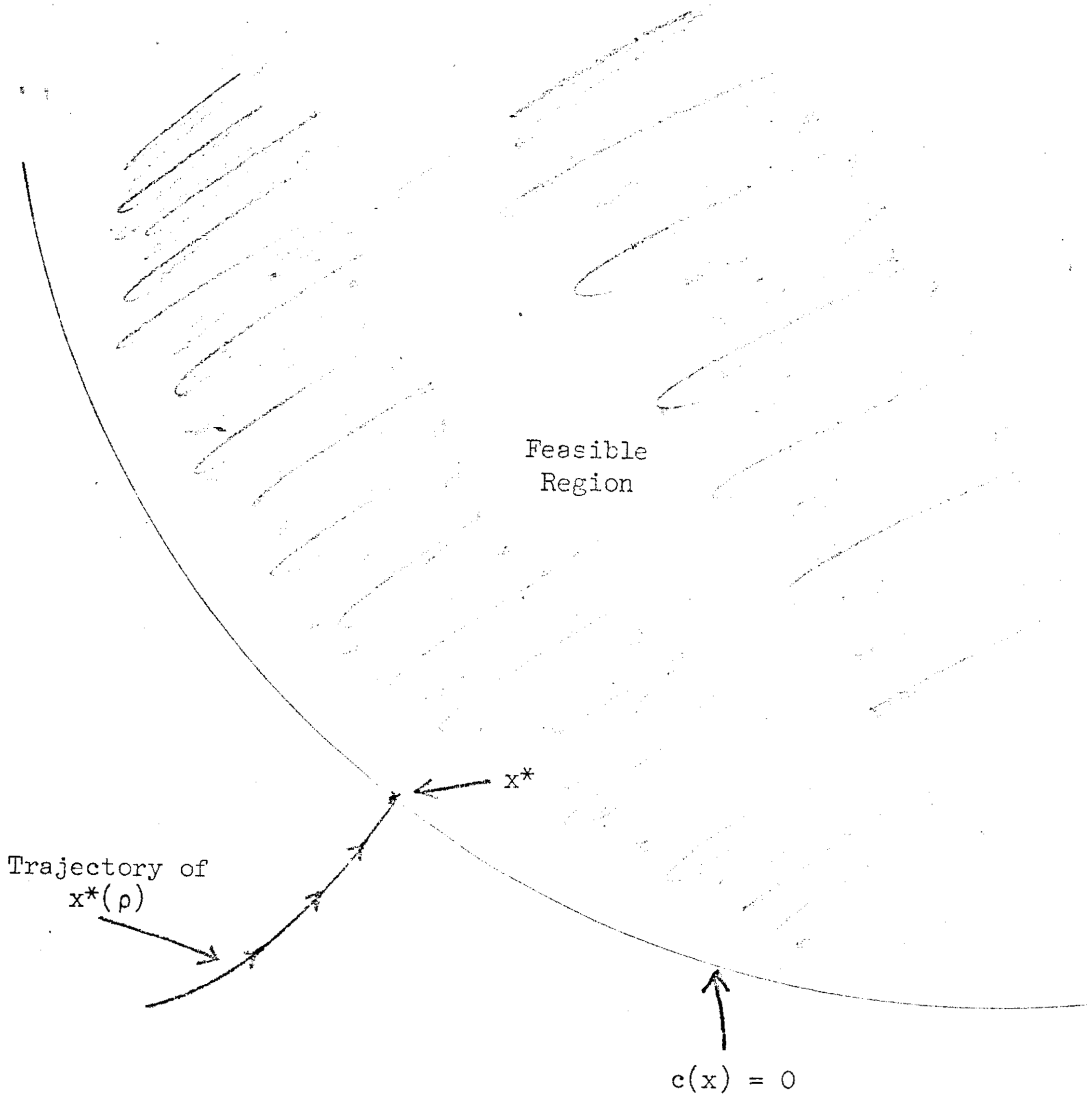


FIGURE 2.3

$$0 = g + \frac{1}{\rho} Gy - A\lambda^* - \frac{1}{\rho} \sum_{i=1}^{m} \lambda_i^* G_i y - \frac{1}{\rho} AU^T y + O\left(\frac{1}{\rho^2}\right),$$

where the columns of U are $\{u_i\}$, $u_i = \frac{\partial \lambda_i}{\partial x}$. Since $g = A\lambda^*$, after cancelling, multiplying by ρ, and letting $\rho \to \infty$, we obtain:

$$Wy - AU^T y = 0,$$

where $W = G - \sum_{i=1}^{m} \lambda_i^* G_i$, the Hessian of the Lagrangian function.

If W is non-singular, the above becomes:

$$y = W^{-1} A U^T y. \qquad (2.2.2)$$

From the previous derivation, $A^T y = -\lambda^*$; applying A^T to the above equation gives:

$$A^T y = A^T W^{-1} A U^T y = -\lambda^*.$$

If $A^T W^{-1} A$ is non-singular, then

$$U^T y = -(A^T W^{-1} A)^{-1} \lambda^*.$$

By applying $W^{-1}A$, which is of full rank, to both sides, and using (2.2.2), we obtain an expression for y in terms of W, A, and λ^*:

$$y = -W^{-1} A (A^T W^{-1} A)^{-1} \lambda^*. \qquad (2.2.3)$$

2.2.4 Difficulties

Although penalty function methods have some significant advantages, they suffer from several deficiencies as well.

Ill-Conditioning of Hessian

The numerical solution of the unconstrained sub-problems is increasingly more difficult as the penalty parameter becomes large. The causes of this phenomenon have been analyzed by Murray (1969a) and Lootsrna (1969) in terms of the conditioning of the Hessian matrix at the minimum of each successive penalty function.

The Hessian of the penalty function is given by:

$$\nabla^2 P(x,\rho) = G + \sum_{i=1}^{m} \rho c_i G_i + \rho A A^T .$$

At $x^*(\rho)$, for large ρ, the first two terms form an approximation to the Hessian matrix of the Lagrangian function at x^*, since $\lim_{\rho \to \infty} \rho c(x^*(\rho)) = -\lambda^*$. However, the matrix $\nabla^2 P$ is dominated for large ρ by the term $\rho A A^T$, and hence approaches a rank-deficient matrix, with rank equal to m, the number of active constraints at x^*.

Expressions can be given for the eigenvalues and eigenvectors of $\nabla^2 P(x^*(\rho),\rho)$ as $\rho \to \infty$ (see Murray, 1971), which reveal that (n-m) eigenvalues are bounded, and have eigenvectors that in the limit lie in the null space of $A(x^*)$; however, there are m eigenvalues of order ρ, i.e., unbounded in the limit, whose corresponding eigenvectors lie in the range of $A(x^*)$.

The progressively worse conditioning of the Hessian matrix of $P(x^*(\rho),\rho)$ will cause numerical difficulties for the methods used to solve each unconstrained sub-problem, and affect the theoretical rate of convergence of the unconstrained algorithm.

Deviation from Original Problem

A further unsatisfactory feature of penalty function methods is illustrated by assuming that the original starting point is very close to the desired solution, x^*. For a typical initial value of ρ (often chosen to be unity), the subsequent estimates of $x^*(\rho)$ would tend to move away from even a neighborhood of x^*. The efficiency of a penalty function method is thus highly dependent on the choice of ρ. Although a penalty function method provides criteria for measuring success at each iteration, the concentration on the sub-problem creates a diversion from the solution of the original constrained problem, because the sub-problem solutions approach x^* only in the limit.

2.2.5 Extension to Inequality Constraints

A penalty function method can be used to solve the inequality constrained problem:

P2: minimize $F(x)$

subject to $c_i(x) \geq 0$, $i=1,2,\ldots,\ell$.

By analogy to the equality case, the functions to be minimized should include a penalty for infeasibility. The quadratic penalty function is used as before, but only the violated constraints are included in the penalty term. The penalty function for inequalities may be written as:

$$P(x,\rho) = F + \frac{\rho}{2} \sum^{\ell} (\min(0,c_i))^2$$

A more compact notation, which will be used throughout this section, is:

$$P(x,\rho) = F + \frac{\rho}{2} \hat{c}^T\hat{c}.$$

where it is assumed that m constraints are currently violated, arid $\hat{c}$ is a vector of length m correspcnding only to the violated constraints. For simplicity, the elements of $\hat{c}$ will be numbered as $\hat{c}_i$, $i=1,2,\ldots,m$, where $\hat{c}_i$ should be taken to mean the ith currently violated constraint. A similar notation will be used for the matrix $\hat{A}$, whose columns are given by $\{\hat{a}_i\}$, $i=1,2,\ldots,m$, the gradients of the constraints in e.

The convergence cf a penalty function method applied to problem P2 can be proved under the same conditions as for problem P1, with the same restriction that the desired local minima of $P(x,\rho)$ exist in a compact domain including x^*. The existence of a trajectory $x^*(p)$, with a multiplier function ?,(ρ), can also be demonstrated under conditions similar to those for the equality problem, with the further restriction that $\lambda_i^* > 0$.

For ρ sufficiently large, the set of constraints violated at $x^*(\rho)$ is equivalent to the set of constraints active at x^*, since the constraints inactive at x^* are strictly feasible in a neighborhood of x^* by continuity. Hence, the minima of successive penalty functions occur, for sufficiently large ρ, at points strictly -infeasible with respect to the active constraints at x^*. Since the condition:

$$g = - \rho\hat{A}\hat{c}$$

holds at $x^*(\rho)$, and $g(x^*) = \hat{A}(x^*)\lambda^*$, where $\hat{A}(x^*)$ denotes the matrix of $\{a_i\}$ corresponding to active ccnstraints, it follows that for $\hat{A}(x^*)$ of full rank, $\lim\limits_{p \to \infty} \rho\, \hat{c}_i = - \lambda_i^* < 0$, and therefore $\hat{c}_i < 0$.

The results given for the penalty function methods applied to equalities - concerning the non-tangential approach of the trajectory $x^*(\rho)$, and the expression for $\lim\limits_{r \to 0} \frac{dx^*(r)}{dr}$ - are identical for the inequality problem, including only the constraints active at x^*, so that $\hat{A}$ and $\hat{c}$ appear in all the corresponding relations for inequalities. The difficulties of penalty function methods remain the same for the inequality problem.

2.3 Barrier Functions - Introduction

Barrier function methods are used to solve the inequality constrained problem:

P2: minimize $F(x)$

subject to $c_i(x) \geq 0$, $i=1,2,\ldots,\ell$.

To apply the barrier function approach to P2, the region for which all constraints are strictly satisfied must have an interior. This restriction means that barrier function methods cannot be used for equality constrained problems. Although an equality constraint, $c_1(x) = 0$, can be expressed by specifying two inequalities ($c_1(x) \geq 0$, $-c_i(x) \geq 0$), there exists no point that can simultaneously strictly satisfy both.

In many physical and engineering applications, the constraint functions not only characterize the desired properties of the solution, but also define the region in which the problem statement is meaningful (for example, F(x) or some of the constraint functions may be undefined cutside 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.

The methods to be considered require strict satisfaction of all constraints at the starting point and subsequent iterates, and are sometimes called "feasible" methods. The continued enforcement of feasibility contrasts with penalty function methods for inequslities, where the constrainsd minimm is approached through a sequence of strictly infeasible points with respect to the active constraints, and feasibility is enforced only in ·ne limit.

As in the penalty function case, 2 barrier function method creates a sequence of modified functions whose successive unconstrained minima should converge in the limit to the constrained solution. Since F(x) will, in general, be minimized at an infeasible point, and the successive estimates are required to be feasible, the modified objective function for a feasible algorithm includes a term to prevent successive iterates from becoming infeasible. If a "barrier" is in effect created at the boundary of the feasible region, by constructing a continuous function with a positive singularity (the "barrier"), any unconstrained minimum of the modified function must lie strictly inside the feasible region. if the weight assigned to the barrier term is decreased toward zero in successive modified functions, the sequence of unconstrained minima should generate a strictly feasible approach to the constrained minimum.

2.4 Barrier Functions - Theory

2.4.1 Definition

A "barrier function" can be defined with wide generality, but only a particular barrier function will be considered here:

$$B(x.r) = F(x) - r \overset{\smile}{\Sigma} \ \ell n \ (c.(x)).$$

called the "logarithmic" 'carrier function (Frisch, 1955). The other most commonly used barrier function is the "inverse" barrier function (Carroll, 1961), given by:

$$B(x,r) = F(x) + r \sum_{i=1}^{\ell} \frac{1}{c_i(x)} .$$

The weighting factor, r, which will approach zero to allow the minima of $B(x,r)$ to converge to x^*, will be called the "barrier parameter".

2.4.2 Description of Algorithm

A barrier function method for solving an inequality constrained problem has the following form:

(Choose a starting point, called $x^*(r^{(0)})$, suck that $c_i(x^*(r^{(0)})) > 0$ for all i,· $i = 1,2,\ldots,\ell$, and an

```
        initial value of r, say $r^{(1)}$;
            $i \leftarrow 1$; ⟩
while (the conditions for a constrained minimum at $x^*(r^{(i-1)})$ are
not satisfied)
repeat ⟨
        Find $x^*(r^{(i)})$, the Unconstrained minimum of $B(x,r^{(i)})$,
starting at $x^*(r(i-1))$;
        Choose $r^{(i+1)} < r^{(i)}$.
        $i \leftarrow i + 1$;
⟩.
```

The use of such a barrier function algorithm involves the unconstrained minimization of a particular function, and thus at each iteration the computation is guided by the effort to solve the sub-problem. As in the penalty function case, an additional decision must be made concerning the choice and alteration of the barrier parameter.

Figure 2.4 illustrates the convergence of the feasible sequence $x^*(r)$ to a constrained minimum.

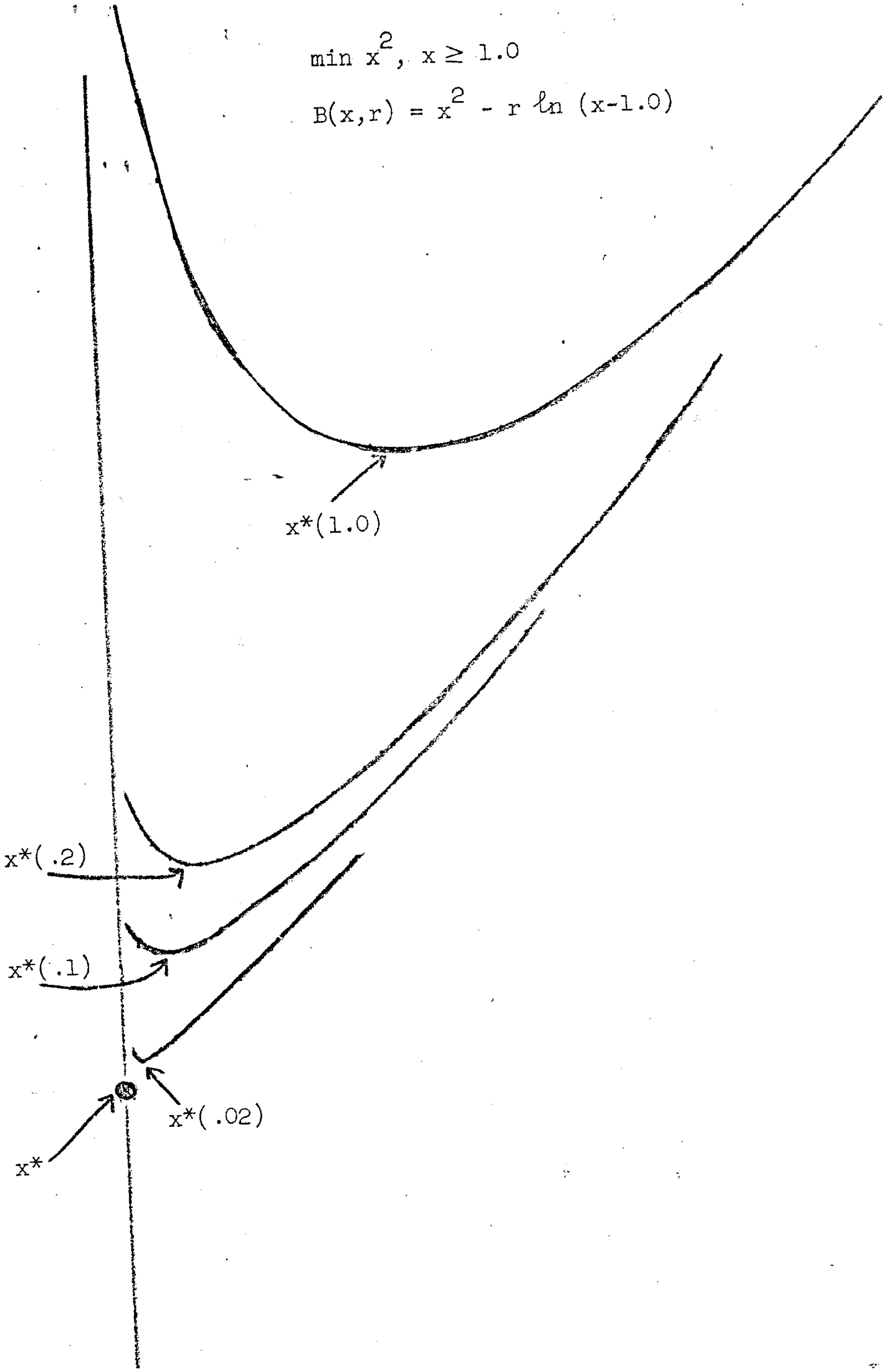
2.4.3 Properties

Convergence

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 minima of successive $B(x,r)$, converges to x^* as $r \rightarrow 0$. As in the analogous proof for penalty function methods, the conditions for convergence do not require ’chat the constraint qualification holds at the limit point, so that barrier function methods will converge to minima not satis-

$\min x^2,\ x \geq 1.0$

$B(x,r) = x^2 - r \ln (x-1.0)$



fying the Kuhn-Tucker conditions.

However, the conditions under which a barrier function method will converge require 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 the convergence proof guarantees convergence of $x^*(r)$ to x^* only within a compact set including x^*, it is possible for the logarithmic barrier function to be unbounded below. This possibility is illustrated in an example given by Powell (1972):

minimize $\dfrac{-1}{x^2+1}$

subject to $x \geq 1$.

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

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

and can decrease without bound. However, this problem is much less likely to happen than with penalty functions, because the feasible region is often closed.

Properties of successive $x^*(r)$

The minima of successive barrier functions exhibit the following 'properties, where $x^{(k)}$ denotes $x^*(r^{(k)})$, $F^{(k)}$ denotes $F(x^*(r^{(k)}))$, and $c_i(k)$ denotes $c_i(x^*(r(k)))$:

(1) $B(x^{(k)},r^{(k)})$, $k = 1,2,\ldots$, is strictly decreasing for sufficiently small $r^{(k)}$ and bounded $c_i^{(k)}$

Proof: By definition

$$B(x^{(k)},r^{(k)}) \leq B(x^{(k-1)},r^{(k)}).$$

Since $\{c_i^{(k)}\}$ are bounded above, for sufficiently small $r^{(k-1)}$, the term $B(x^{(k-1)}, r^{(k)})$ can be bounded as follows:

$$F^{(k-1)} - r^{(k)} \sum_{i=1}^{\ell} \ln c_i^{(k-1)} \leq F^{(k-1)} - r^{(k-1)} \sum_{i=1}^{\ell} \ln c_i^{(k-1)} =$$

$$B(x^{(k-1)}, r^{(k-1)}),$$

because $0 < r^{(k)} < r^{(k-1)}$.

(2) $F(x^{(k)})$, $k = 1,2,\ldots,$ is nonincreasing.

Proof: By definition of $x^*(r^{(k)})$,

$$\text{(a)} \quad F^{(k)} - r^{(k)} \sum_{i=1}^{\ell} \ln c_i^{(k)} \leq F^{(k-1)} - r^{(k)} \sum_{i=1}^{\ell} \ln c_i^{(k-1)}, \text{ and}$$

$$\text{(b)} \quad F^{(k-1)} - r^{(k-1)} \sum_{i=1}^{\ell} \ln c_i^{(k-1)} \leq F^{(k)} - r^{(k-1)} \sum_{i=1}^{\ell} \ln c_i^{(k)}.$$

Multiplying (a) by the factor $\frac{r^{(k-1)}}{r^{(k)}} > 1$, and adding to (b), yields after re-arranging:

$$\left(\frac{r^{(k-1)}}{r^{(k)}} - 1\right) F^{(k)} \leq \left(\frac{r^{(k-1)}}{r^{(k)}}\right) F^{(k-1)}, \quad \text{or}$$

$$F^{(k)} \leq F^{(k-1)}.$$

(3) $-\sum_{i=1}^{\ell} \ln c_i^{(k)}$, $k = 1,2,\ldots$ is nondecreasing.

Proof: Combining (b) and the result $F_{(k)} \leq F_{(k-1)}$, we obtain:

$$-\sum^{\ell} \ln c_i^{(k-1)} \leq -\sum^{\ell} \ln c_i^{(k)}.$$

It should be noted that property (3) does not imply that the constraint values decrease at successive $x^*(r)$. A reduction in the barrier parameter only allows the constraints to approach the boundary of the feasible region, but does not enforce any decrease.

Lagrange Multiplier Estimates

At the minimum of $B(x,r)$:

$$\nabla B(x,r) = g - r\sum_{i=1}^{\ell} \frac{1}{c_i} a_i = g - A \begin{bmatrix} \frac{r}{c_1} \\ \vdots \\ \frac{r}{c_\ell} \end{bmatrix} = 0,$$

so that the gradient of $F(x)$ is a non-negative linear combination of the constraint gradients, where the coefficient of a_i is $\overline{c_i}$. As $x^*(r)$ approaches x^*, the expression $\frac{r}{c_i}$ will go to zero as $r \to 0$ if c_i is not active at x^*, since c_i will be bounded strictly away from zero in a neighborhood of x^*. Hence, for sufficiently small r, the relation holding at $x^*(r)$ can be written:

$$g = \hat{A} \begin{bmatrix} \frac{r}{\hat{c}_1} \\ \vdots \\ \frac{r}{\hat{c}_m} \end{bmatrix} + O(r).$$

It is assumed throughout that m constraints are active at x^*. The vector $\hat{c}$, of length m, contains the active constraints, with elements numbered for simplicity as $\hat{c}_1, \ldots, \hat{c}_m$, where $\hat{c}_i$ should be interpreted as the ith active constraint. Similarly, the matrix $\hat{A}$ contains only the m gradients of constraints active at x^*.

The quantities $\lambda_i(x^*(r)) = \lambda_i(r) = \overline{\hat{c}_i}$, defined only for the active constraints, therefore satisfy a relationship with g arid $\hat{A}$ analogous to the multiplier relation that must hold at x^* if

$\hat{A}(x^*)$ has full rank:

$$g(x^*) = \hat{A}(x^*)\lambda^*.$$

By continuity, for sufficiently small r, the vector $\lambda(r)$ is an increasingly accurate approximation to λ^*, the vector of Lagrange multipliers at x^*.

Trajectory of $x^*(r)$

Under suitable assumptions, 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). If:

(1) F and $\{c_i\}$ are twice differentiable;

(2) $\hat{A}(x^*)$ has full rank;

(3) $\lambda_i^* > 0$, $i = 1,2,\ldots,m$ (all Lagrange multipliers strictly positive);

(4) the sufficient conditions given in Section 1.2 for a constrained minimum of P2 are satisfied at x^*;

Then for sufficiently small r, there exists a continuously differentiable trajectory $x^*(r)$ such that $\lim\limits_{r \to 0} x^*(r) = x^*$, and for any $r > 0$, $x^*(r)$ is a local minimum of $B(x,r)$. As with penalty functions, it is possible to extrapolate along the trajectory, but this topic will not be discussed here.

The trajectory defined by $x^*(r)$ has several interesting properties. Expanding about x^* gives the following expression for $x^*(r)$:

where $y = \lim_{r \to 0} \frac{x^*(r) - x^*}{r} = \lim_{r \to 0} \frac{dx^*(r)}{dr}$.

At $x^*(r)$,

$$\nabla B(x,r) = g - A \begin{bmatrix} \frac{r}{c_1} \\ \vdots \\ \vdots \\ \frac{r}{c_\ell} \end{bmatrix} \equiv 0 .$$

Differentiating this identity with respect to r yields:

$$-A \begin{bmatrix} \frac{1}{c_1} \\ \cdot \\ \\ \vdots \\ \frac{1}{c_\ell} \end{bmatrix} + (G - \sum_{i=1}^{\ell} \frac{r}{c_i} G_i + A \begin{bmatrix} \frac{r}{c_1^2} & 0 \ldots & & \\ & \frac{r}{c_2^2} & & \\ 0 & & \ddots & \\ \vdots & & & \frac{r}{c_\ell^2} \end{bmatrix} A^T) \frac{dx^*(r)}{ar} = C.$$

Multiplying through by r, and re-arranging, we obtain:

$$(rG - \sum_{i=1}^{\ell} \frac{r^2}{c_i} G_i + A \begin{bmatrix} \frac{r^2}{c_1^2} & & & \\ & \frac{r^2}{c_2^2} & & \\ & & \ddots & \\ & & & \frac{r^2}{c_\ell^2} \end{bmatrix} A^T) \frac{dx^*(r)}{dr} \qquad \begin{bmatrix} \frac{r}{c_1} \\ \cdot \\ \\ \vdots \\ \frac{r}{c_\ell} \end{bmatrix} \tag{2.4.1}$$

For r sufficiently small, the inactive constraints are bounded away from zero, and the value $\frac{r}{c_i}$ is $O(r)$. If the Lagrange multipliers at x^* are bounded, for r sufficiently small the quantity $\lambda_i(r) = \frac{r}{c_i}$ is also bounded, since $\lim_{r \to 0} \frac{r}{c_i} = \lambda_i^*$. As r approaches zero, the only significant term on the left-hand side of (2.4.1) is the matrix

$$A \begin{bmatrix} \frac{r^2}{c_1^2} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \cdot \frac{r^2}{c_\ell^2} \end{bmatrix} A^T,$$

since all other matrices are bounded in norm by a multiple of r. The columns of A corresponding to inactive constraints make an $\mathcal{O}(r^2)$ contribution to the elements of the matrix on the left-hand side, and an $\mathcal{O}(r)$ contribution to the vector on the right-hand side. Hence we obtain:

$$A \begin{bmatrix} \frac{r^2}{\hat{c}_1^2} & & & \\ & \frac{r^2}{\hat{c}_2^2} & & \\ & & \cdot & \\ & & & \cdot \frac{r^2}{\hat{c}_m^2} \end{bmatrix} \hat{A}^T \frac{dx^*(r)}{dr} = \hat{A} \begin{bmatrix} \frac{r}{\hat{c}_1} \\ \vdots \\ \vdots \\ \frac{r}{\hat{c}_m} \end{bmatrix} + \mathcal{O}(r).$$

If $\hat{A}(x^*)$ has full rank, then for sufficiently small r, $\hat{A}(x^*(r))$ will also be of full rank. Cancelling $\hat{A}$, ignoring the $\mathcal{O}(r)$ term, and dividing the <u>ith</u> row by $\frac{r^2}{\hat{c}_i^2}$ gives

$$\hat{A}^T \frac{dx^*(r)}{dr} = \begin{bmatrix} \frac{\hat{c}_1}{r} \\ \\ \\ \frac{\hat{c}_m}{r} \end{bmatrix}.$$

Since $\frac{r}{\hat{c}_i} \to \lambda_i^*$ as $r \to 0$, the final result is

$$\hat{A}^T \lim_{r \to 0} \frac{dx^*(r)}{dr} \equiv \hat{A}^T y = \begin{bmatrix} \frac{1}{\lambda_1^*} \\ \vdots \\ \frac{1}{\lambda_m^*} \end{bmatrix} \qquad (2.4.2)$$

An alternative derivation of this result can be obtained from the relationship that holds for the active constraints:

$$\frac{r}{\hat{c}_i(x^*(r))} = \lambda_i(x^*(r)), \text{ or}$$

$$r = \hat{c}_i(x^*(r))\, \lambda_i(x^*(r)), \quad i = 1,2,\ldots,m.$$

Expanding about x^* gives the following, where all functions are evaluated at x^*, and $u_i = \frac{\partial \lambda_i}{\partial x}$:

$$r = (\hat{c}_i + r\hat{a}_i^T y + O(r^2))\,(\lambda_i^* + r u_i^T y + O(r^2)).$$

Noting that $\hat{c}_i = 0$, and dividing through by r, yields:

$$1 = \lambda_i^* \hat{a}_i^T y + O(r).$$

ktting $r \to 0$, for $\lambda_i^* \neq 0$, the result is as before:

$$\hat{a}_i^T y = \frac{1}{\lambda_i^*}, \quad i = 1,2,\ldots,m.$$

The relationship (2.4.2) implies that the minima of successive barrier functions do not approach x^* along any tangent to a constraiot, for $\lambda_i^* \neq 0$, $|\lambda_i^*| < \infty$, and thus the use of a linear approximation to the behavior of an active constraint along this trajectory is justified. Figure 2.5 displays the typical approach

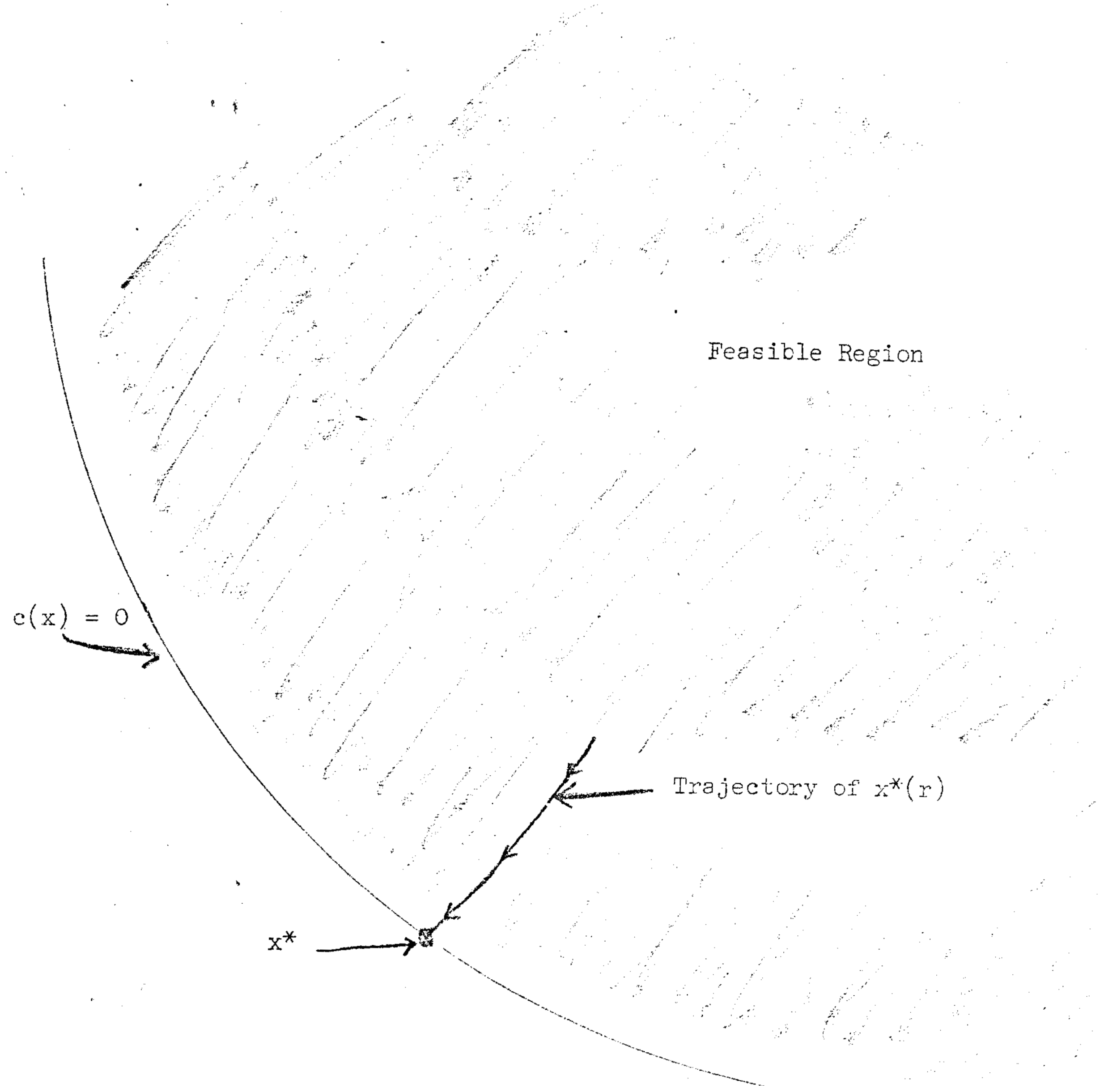


FIGURE 2.5

of a barrier function trajectory to x*. This property will be discussed in more detail in Chapter 5, along with considersticn of how it may be exploited in the design of a feasible algorithm.

An expression for $y = \lim_{r \to 0} \frac{dx^*(r)}{dr}$ can be obtained from the identity:

$$0 = \nabla B(x^*(r),r) \tag{2.4.3}$$

$$= g(x^*(r)) - A(x^*(r)) \begin{bmatrix} \frac{r}{c_1(x^*(r))} \\ \frac{r}{c_2(x^*(r))} \\ \vdots \\ \frac{r}{c_\ell(x^*(r))} \end{bmatrix} .$$

Let the vector function $\mu(x^*(r))$ be defined 'cy:

$$\mu_i(x^*(r)) = \frac{r}{c_i(x^*(r))} , \quad i = 1,2,\ldots,\ell.$$

in the following, the set I denotes the set of indices of the active constraints; if $i \in I$, $c_i(x^*) = 0$, and if $i \notin I$, $c_i(x^*)$ is bounded away from zero.

<u>Case 1:</u> The ith constraint is active at x* ($i \in I$).

For an active constraint, $\mu_i(x^*(r))$ is equivalent 'io the appropriate Lagrange multiplier approximation, λ_i, defined earlier, which is known to be differentiable along the trajectory x*(r), and it is possible to expand $\mu_i(x^*(r))$ around x*. Let $\bar{i}$ be the index in the vector $\hat{c}$ of active constraints (or, equivalently, in I) corresponding to c_i; then:

$$\mu_i(x^*(r)) = \lambda_i^* + r\, u_i^T y + O(r^2), \text{ where } u_i = \frac{\partial \hat{\lambda}_i}{\partial x} ;$$

Case 2: The ith constraint is inactive at x^* ($i \notin I$).

For an inactive constraint, $\mu_i(x^*(r))$ is differentiable because of the assumed properties of c_i; in addition, μ_i and all its derivatives are of order r. It is not possible to obtain an accurate expansion of $\mu_i(x^*(r))$ around x^*, since μ_i and all its derivatives are zero when $r = 0$.

The identity (2.4.3) can be written:

$$g(x^*(r)) - A(x^*(r))\mu(x^*(r)) = 0.$$

Expanding g and the elements of μ and A corresponding to active constraints, we obtain the following:

$$g + rGy - \hat{A}\lambda^* - r\sum_{i=1}^{m} \lambda_i^* \hat{G}_i y - r\hat{A}U^T y - \sum_{i \notin I} \frac{r}{c_i(x^*(r))} a_i(x^*(r)) + O(r^2) = 0,$$

where all functions are evaluated at x^* unless otherwise specified, $\hat{G}_i$ denotes the Hessian corresponding to i' and the columns. of U are given by $\{u_i\}$.

At x^*, $g = \hat{A}\lambda^*$; cancelling and dividing by r gives:

$$(G - \sum_{i=1}^{m} \lambda_i^* \hat{G}_i)y = \hat{A}U^T y + \sum_{i \notin I} \frac{1}{c_i(x^*(r))} a_i(x^*(r)) + O(r).$$

Taking the limit as $r \to 0$ gives:

$$Wy = \hat{A}U^T y + t.$$

where $W = G - \sum_{i=1}^{m} \lambda_i^* \hat{G}_i$, and the vector **t** is defined by:

$$t = \sum_{i \notin I} \frac{1}{c_i(x^*)} a_i(x^*).$$

If W is non-singular, the result is:

$$y = W^{-1}\hat{A}U^T y + W^{-1} t . \qquad (2.4.4)$$

Since $\hat{A}^T y = \begin{bmatrix} \frac{1}{\lambda_1^*} \\ \vdots \\ \frac{1}{\lambda_m^*} \end{bmatrix}$, if $\hat{A}^T$ is applied

to both sides of (2.4.4), the relationship becomes:

$$\hat{A}^T y = \hat{A}^T W^{-1} \hat{A} U^T y + \hat{A}^T W^{-1} t = \begin{bmatrix} \frac{1}{\lambda_1^*} \\ \vdots \\ \frac{1}{\lambda_m^*} \end{bmatrix} .$$

$\hat{A}^T W^{-1} \hat{A}$ is nonsingular if $\hat{A}$ is of full rank, giving:

$$U^T y = (\hat{A}^T W^{-1} \hat{A})^{-1} \begin{bmatrix} \frac{1}{\lambda_1^*} \\ \vdots \\ \frac{1}{\lambda_m^*} \end{bmatrix} - (\hat{A}^T W^{-1} \hat{A})^{-1} \hat{A}^T W^{-1} t .$$

The desired result is ottained by applying $W^{-1}\hat{A}$ to the above, and then using the previously given relationship (2.4.4):

$$y = W^{-1}\hat{A} \, (\hat{A}^T W^{-1} \hat{A})^{-1} \begin{bmatrix} \frac{1}{\lambda_1^*} \\ \vdots \\ \frac{1}{\lambda_m^*} \end{bmatrix} - W^{-1}(\hat{A}(\hat{A}^T W^{-1} \hat{A})^{-1} \hat{A}^T W^{-1} - I)t . \quad (2.4.5)$$

2.4.4 Difficulties

Ill-conditioning of Hessian

The unconstrained minimization problems to be solved by a barrier function method become more difficult as the iterates converge to x^*, if there are any constraints active at x^*, due to the increasingly poor conditioning of the Hessian matrix $\nabla^2 B(x^*(r),r)$. 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 $B(x,r)$ is given by:

$$\nabla^2 B(x,r) = G - \sum_{i=1}^{\ell} \frac{r}{c_i} G_i + A \begin{bmatrix} \frac{r}{c_1^2} & & & \\ & \frac{r}{c_2^2} & & \\ & & \ddots & \\ & & & \frac{r}{c_\ell^2} \end{bmatrix} A^T.$$

For inactive constraints, c_i is bounded away from zero, and thus the quantities $\frac{r}{c_i}$ and $\frac{1}{c_i^2}$ go to zero as $x^*(r)$ approaches x^*. For the active constraints, $\frac{r}{c_i(x^*(r))}$ approaches the corresponding Lagrange multiplier, and hence for non-zero Lagrange multipliers the ratio $\frac{r}{c_i^2(x^*(r))}$ is unbounded as $c_i \to 0$. The first two terms of $\nabla^2 B(x^*(r),r)$ constitute an increasingly accurate approximation to $W(x^*)$, the Hessian of the Lagrangian function; but $\nabla^2 B(x^*(r),r)$ approaches a rank-deficient matrix of rank m, the number of active constraints at x^*. It has been shown (see Murray, 1971) that $\nabla^2 B(x^*(r),r)$ has m unbounded eigenvalues as $r \to 0$, with corresponding eigenvectors in the range of $\hat{A}$; the remaining (n-m) eigenvalues are bounded and their eigcuvectors lie in the null space of $\hat{A}$.

Linear Search

A further difficulty in applying the usual unconstrained techniques to minimize barrier functions is that the one-dimensional search procedures, based on polynomial approximation of the function to be minimized, are often inefficient. This topic is discussed in detail in Chapter 3.

Deviation from Original Problem

As with penalty function methods, the sub-problem solutions for berrier functions converge to x^* only in the limit, because x^* is not the minimum of $B(x,r)$ for any non-zero r. Hence the structure of the sequence of unconstrained minimizations may obscure the properties of the original constrained problem. The requirement of maintaining feasibility in a reasonable way adds further complications to a barrier function algorithm.

2.5 Summary

Both penalty and barrier function methods are effective in solving nonlinearly constrained problems, but their design contains notable inherent defects. Tne original hope for these methods was that an unconstrained minimization technique could be applied to the modified functions with the same success as for any other function. However: the difficulties discussed have made this hope unrealizable. The problems of ill-cciditioning impair the convergence of unconstrained methods for both penalty and barrier functions methods. For barrier functions, any unconstrained algorithm must undergo non-trivial modifications in order to maintain feasibility.

Many recent algorithms have been devised in an effort to overcome the deficiencies of penalty and barrier function methods - for example, to deal more directly with properties of the constrained solution, and to avoid constructing a sequence of sub-problems that converges only in the limit. The motivation and structure of a large class of alternative methods, based on the Lagrangian function, are presented in Chapter 4.

CHAPTER 3. Linear Searches to be used with the Logarithmic Barrier Function

CHAPTER 3

LINEAR SEARCHES TO BE USED WITH THE LOGARITHMIC BARRIER FUNCTION

3.1 General Discussion of Linear Searches; Inadequacy of Usual Strategy for Minimizing Barrier Functions

The model minimization algorithm requires at each iteration the determination of a positive step, α, along the given search direction, p. In early algorithms, the step length α was chosen so that $x + \alpha p$ was a highly accurate approximation to the minimum of $F(x)$ along p. In general, this one-dimensional minimization sub-problem cannot be solved explicitly, nor the step to the minimum located exactly. Hence, an iterative procedure is required, with termination criteria that will be satisfied by a sufficiently accurate estimate of the minimum.

It is no longer Considered necessary or efficient to locate a highly accurate estimate of the minimum of $F(x)$ along the search direction at every iteration of most minimization algorithms. However, the same iterative procedure that would be used to find an accurate minimum can be followed to determine an "acceptable" step along p, which satisfies less stringent termination criteria. The various termination criteria that might be specified are not directly relevant to the present discussion, which will be concerned with the procedure by which successive estimates of the solution are obtained. The teras "one-dimensional minimization" and "linear search" will be used to denote this iterative procedure, which remains the same for whatever termination criteria are given.

It will be assumed that the function whose (approximate) minimum is sought is decreasing along the specified search direction at the initial point, or, equivalently, that any search direction is guaranteed to be a descent direction for the given function. This assumption means that the step α to the (approximate) minimum is positive. The iterative methods for locating an acceptable α determine an initial step, $\alpha^{(0)}$, and successively compute $\alpha^{(1)}$, $\alpha^{(2)}$, etc., until $x + \alpha^{(k)}p$ satisfies the given termination criteria; since α is known to be positive, all estimates $\alpha^{(k)}$ are required to be positive as well.

The task of developing an efficient and reliable algorithm for one-dimensional minimization is extremely difficult. There is a large literature on this subject, and existing methods continue to be improved (see Gill and Murray, 1974a; Brent, 1973). The characteristics that a iinear search algorithm should possess include: rapid convergence en well-behaved functions; guaranteed convergence for any function satisfying certain weak assumptions; numerical stability; and finite termination for any function. These qualities may be incompatible, in that techniques to achieve guaranteed convergence and numerical stability may impede possible fast convergence. In software designed for general use, the goals of guaranteed convergence and stability will normally take precedence over that of rapid convergence if these aims conflict.

The approach taken in the best available iinear search algorithms is to combine safeguards to guarantee convergence and stability with a procedure for polynomial interpolation or extrapolation. A well-behaved function that satisfies the assumption of sufficient smoothness can be approximated in a neighborhood by the low-order terms of its Taylor series, and hence in one dimension by a quadratic or cubic polynomial. Most linear search algorithms fit either a parabola or cubic to known values of the function and its derivatives along the search direction. The coefficients of the fitted polynomial solve a system of linear equations;

the minimum of the fitted polynomial can thus be computed explicitly, to be taken as the next estimate of the minimum of the general function if appropriate safeguarding checks are satisfied. This procedure is reasonable, and highly successful, for most functions to be minimized by unccnstrained methods.

However, the usual technique of approximation by quadratic cr cubic polynomials suffers from certain defects when used to minimize barrier functions, which, although continucus and smooth in the strictly feasible region, are not "well-behaved" in the normal sense. Their defining characteristic is singularity at the boundary of the feasible region. Unlike penalty functions, whose large derivatives are due to a parameter under external control, a barrier function must contain a singularity; and a continuous function with a singularity will have by definition large and rapldly varying derivatives. In the case of the logarithmic barrier function, the first derivatives go to infinity as $1/c$ as c approaches zero, and the seccnd derivatives behave like $1/c^2$. Hence, the hope of a good approximation to a 'carrier function by neglecting the higher derivatives will not be fulfilled in the vicinity of the boundary of the feasible region. Figure 3.1 displays the cne-dimensional variation of a logarithmic barrier function as the boundary is approached.

Although an adequately safeguarded linear search based on polynomial fitting will. be able, with proper care, to locate the approximate minimum of a barrier function accurately, the number of constraint and function evalusticns required to converge may be excessive. The predicted minima of the fitted polynomials often lie in the infeasible region; and even within the feasible region, the fitted polynomials tend to be a poor approximation to the barrier function. The inadequacy of the usual linear search procedures for barrier functions has been discussed by

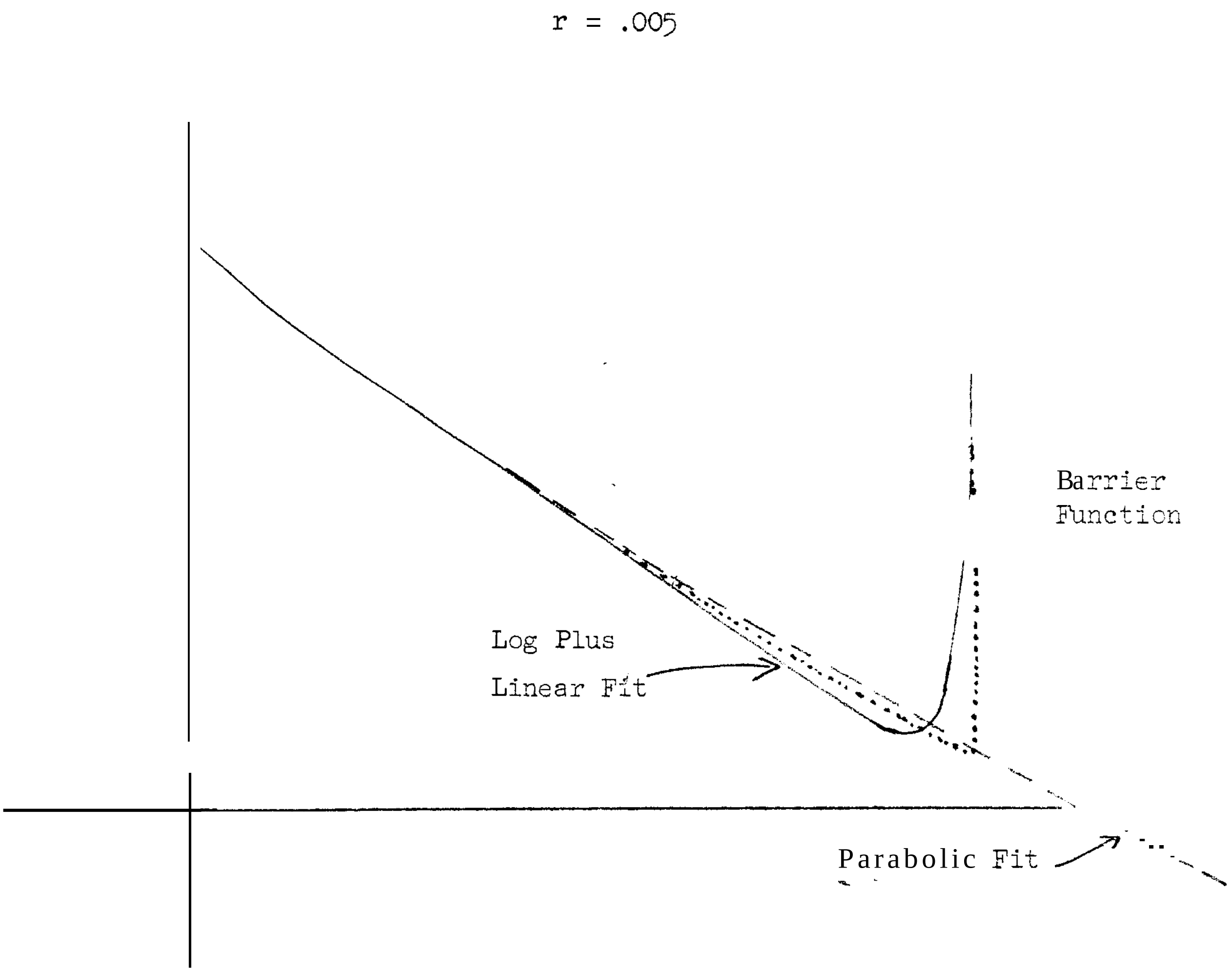


FIGURE 3.1

several authors (see Fletcher and McCann, 1969 ; Murray, 1969a ; Lasdon et al., 1973 ; Ryan, 1974). Various alternative strategies have been proposed, which involve using an approximating function that will better emulate the behavior of the barrier function. One possibility is to use an approximating function with the same kind of singularity as the particular barrier function; the behavior of the fitted function, including the location of its minimum, should then be similar to that of a barrier function. Consequently, the minimum of such a special function could be taken as the next estimate of the minimum of the barrier function, to be used within the linear search procedure in exactly the same way as the estimated minima of polynomial approximations. However, because the parameters of such a special approximating function will no longer be determined in general by a system of linear equations, it may not be possible to determine explicitly the defining parameters and the minimum cf the fitted function.

3.2 Fitting of Special Functions

3.2.1 Discussion

The special functions to be considered are designed to contain a single logarithmic singularity. This restriction to a particular barrier function contrasts with the general approach taken by Lasdon et al. (1973), where a special function is developed by the application of the form cf the barrier function to linearized approximations to the objective function and the constraints. The approach taken here allows a much simpler solution to the problem, where the known form of the singularity is directly exploited. A single function is used to approximate the behavior of the barrier function along the seerch direction, and it is not necessary to make separate approximations to F and $\{c_i\}$.

The logarithmic barrier function can be evaluated only at feasible points, and is undefined beyond the first singularity along a particular direction. Therefore, only feasible points are considered in fitting the approximating functions, since the data to be used will be values of the barrier function and its gradient.

It will be assumed that we seek an approximation to the minimum along the search direction of $B(x,r) = F - r \sum_{i=1}^{\ell} \ln(c_i)$, where the barrier parameter, r, is known and fixed throughout the linear search. The special functions to be considered are the following, where the parameter θ represents the' one-dimensional variation along the search direction:

(a) a linear function plus a logarithmic singularity, of the form:

$$f_L(\theta) = \hat{a} + \hat{b}\ Q - r \ln(\hat{d} - \theta);$$

(b) a quadratic function plus a logarithmic singularity, of the form:

$$f_Q(\theta) = \hat{a} + \hat{b}\ e + \hat{c}\ \theta^2 - r \ln(\hat{d} - \theta).$$

These formulations attribute all the singular behavior to one location ($\theta = \hat{d}$); in fact, the singularity displayed by a barrier function depends in general upon the first constraint zero encountered along a given direction, and zeros that nay occur beyond the first are irrelevant. This behavior becomes especially marked for small values of the barrier parameter when the constraint functions bounded away from zero have almost no influence on the barrier function, and the nearest constraint has influence only very close to its zero along the given direction. A smooth function with a suitable damped singularity should, therefore: be an excellent model of a barrier function close to the boundary of the feasible region.

3.2.2 Linear Function Plus Logarithmic Singularity

The special function to be fitted is of the form:

$$fL = \hat{a} + \hat{b}\,\theta - r\ln(\hat{d} - \theta).$$

Differentiating f_L with respect to θ gives:

$$f_L' = \hat{b} + \frac{r}{\hat{d} - \theta}, \text{ and}$$

$$f_L'' = \frac{r}{(\hat{d} - \theta)^2} .$$

Since the barrier function is decreasing at $\theta = 0$, this same condition will be required of fL' s@that

$$f_L'(0) = \hat{b} + \frac{r}{\hat{d}} < 0.$$

The singularity, $\hat{d}$, is assumed to be a positive step along the current direction, and therefore, $\hat{b} < 0$.

A stationary point of f_L occurs at θ^* such that $f_L'(\theta^*) = 0$, or

$$\hat{b} + \frac{r}{\hat{d} - \theta^*} = 0, \text{ so that}$$

$$\theta^* = \hat{d} + \frac{r}{\hat{b}} .$$

The expression for θ^* is equal to the value of $f_L'(0)$ times $(\frac{\hat{d}}{\hat{b}})$, and since $(\frac{\hat{d}}{\hat{b}})$ and $f_L'(0)$ are negative, θ^* must be positive. f_L'' is everywhere positive., so that the stationary point must be a minimum of f_L.

In order to specify the function f_L, three indepecdent pieces of information are required about the behavior of the function to be fitted along the search direction; f_L will therefore be used in circumstances where a parabolic fit would normally be carried out to minimize a general function. Figure 3.1 illustrates a barrier function and its approximations by the function f_L and a parabola, both. using the same data. It is clear that the special function gives a more accurate prediction than the parabola of the minimum of the barrier fanction.

The special function f_L will be fitted with the same sets of data used by a typical parabolic line search, namely: (1) one function value, the corresponding gradient, and a second function value; (2) three function values. In the following discussion, f_i will denote the function value at θ_i, and g_i is the corresponding gradient value.

Case 1. Two function values, one gradient

The three unknown parameters of f_L - $\hat{a}$, $\hat{b}$, and $\hat{d}$ - must be solved for in terms of the known values. Assuming that $\theta_2 > \theta_1$, the equations specifying f_L are:

(A) $f_1 = \hat{a} + \hat{b}\,\theta_1 - r \ln (\hat{d}$

(B) $g_1 = \hat{b} + \dfrac{r}{\hat{d} - \theta_1}$

(C) $f_2 = \hat{a} + \hat{b}\,\theta_2 - r \quad (\hat{d} - \theta_2)$.

There is no loss of generality in assuming that $\theta_1 = 0$, so that from (A) and (B) we obtain:

$\hat{a} = f_1 + r \ln \hat{d}$, and

$\hat{b} = g_1 - \frac{r}{\hat{d}}$.

Substituting these values into (C) yields:

$f_2 = f_1 + r \ln \hat{d} + (g_1 - \frac{r}{\hat{d}})\,\theta_2 - r \ln (\hat{d} - \theta_2)$, or,

after re-arranging:

$$\ln \frac{\hat{d} - \theta_2}{\hat{d}} + \frac{\theta_2}{\hat{d}} = \frac{1}{r}\,(\theta_2 g_1 - (f_2 - f_1)) \qquad (3.2.1)$$

found which satisfies (3.2.1), the values $\hat{a}$, $\hat{b}$, and 8” can then be computed from the previously derived relationships.

Consider (3.2.1) as a nonlinear equation in terms of the function

$$\Phi_1(d) = \ln \frac{d - \theta_2}{d} + \frac{\theta_2}{d} + k_1,$$

where $k_1 = \frac{1}{r}(f_2 - f_1 - \theta_2 g_1)$. The problem to be solved then becomes that of finding a solution $\hat{d}$ to satisfy

$$\Phi_1(d) = 0.$$

Because θ_2 is feasible, the value of the singularity, **d**, must lie in the interval (θ_2, ∞). The function $\Phi_{+}(d)$ has the following properties:

$\Phi_1 \to -\infty$ as $d \to \theta_2 +$;

$\Phi_1 \to k_1$ as $d \to \infty$;

$\Phi_1' > 0$ for all finite d;

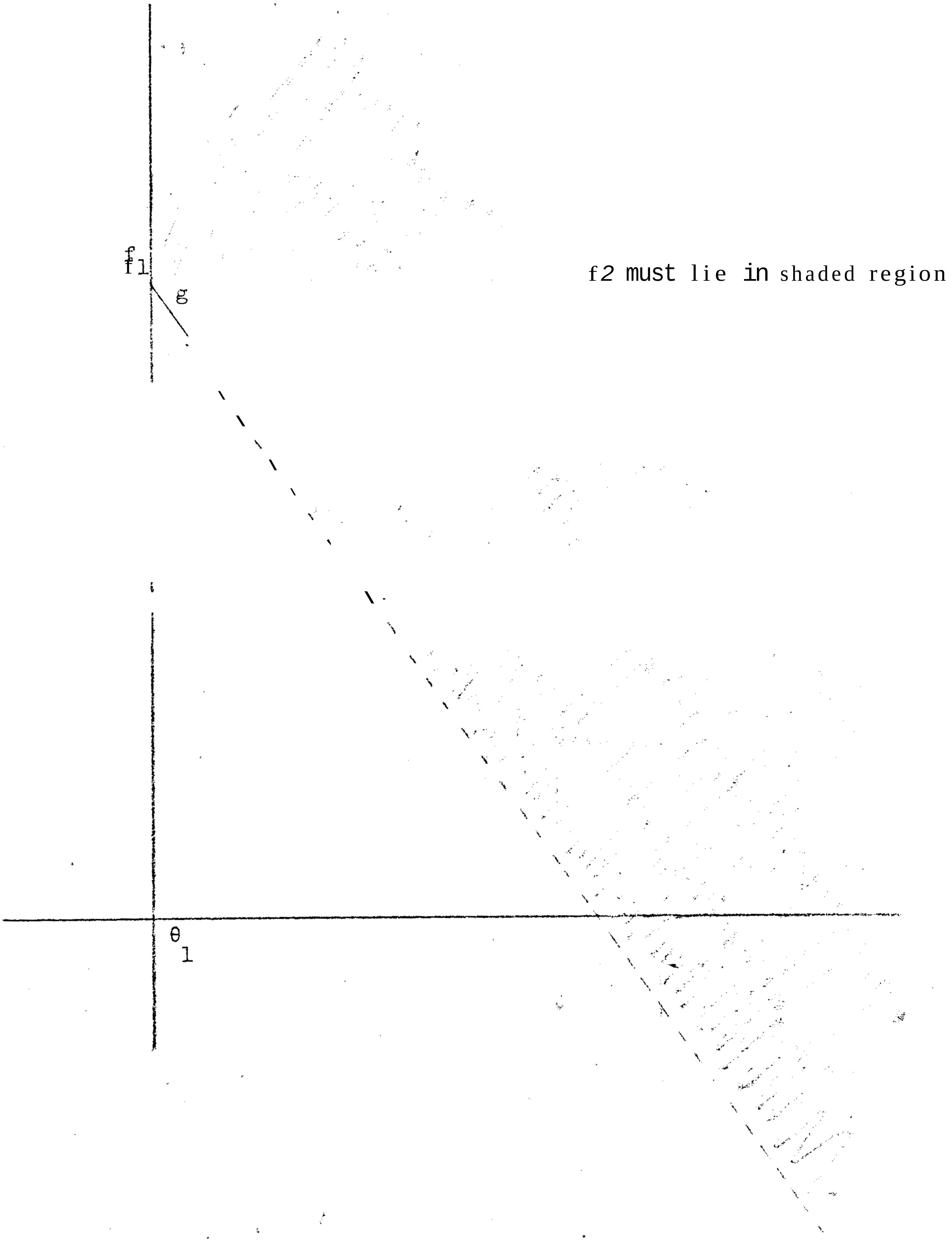
$\Phi_1' \to \infty$ as $d \to \theta_2 +$;

$\Phi_1' \to 0$ as $d \to \infty$; and

$\Phi_1'' < 0$ for all finite d.

These relations imply that there is a unique zero of Φ_1 in (θ_2, ∞), provided that $k_1 > 0$, i.e., $\frac{f_2 - f_1}{\theta_2} > g_1$. This requirement means that the function value at θ_2 must be larger than the linear approximation at θ_1 would have predicted, i.e., f2 must lie in the shaded region of Figure 3.2.

Although the numerical. solution of the equation $\Phi_1(d) = 0$ would yield the desired value $\hat{d}$, Φ_{+} is an ill-behaved function, unsuitable for the usual zero-finding techniques such as Newton's



f_2 must lie in shaded region

FIGURE 3.2

method. Figure 3.3 illustrates the behavior of Φ_1.

An attempt to apply Newton's method in Region I would cause very slow convergence to the zero, although all estimates would undershoot and hence could not diverge. in Region 11, Newton's method might easily yield an estimate to the left of θ_2, and safeguards would need to be incorporated to prevent divergence.

To avoid these difficulties, the problem of solving $\Phi_1(d) = 0$ can be transformed into an equivalent problem that is easy to solve. If we define a new variable $y = \frac{\theta_2}{d}$, so that $0 < y < 1$ for the admissible range of d, the equation $\Phi_1(d)$ can be written in teras of y as

$$\ell n(1-y) = -y-k_1.$$

Taking exponentials of both sides gives the equation

$$1-y = e^{-y} e^{-k_1},$$

and the value $\hat{y}$ which satisfies this equation is a zero of a new function,

$$\Psi_1(y) = 1-y-e^{-y}e^{-k_1}.$$

Differentiating with respect to y gives:

$$\Psi_1' = -1 + e^{-y}e^{-k_1}$$

$$\Psi_1'' = -e^{-y}e^{-k_1},$$

and thus Ψ_1' and Ψ_1'' are negative for all y in the interval (0,1), if $k_1 > 0$.

Furthermore,

$$\Psi_1(0) = 1-e^{-k_1} > 0,$$

$$\Psi_1(1) = -e^{-1}e^{-k_1} < 0,$$

so that Ψ_1 has a unique zero $\hat{y}$ in (0,1) if $k_1 > 0$. Figures 3.4 and

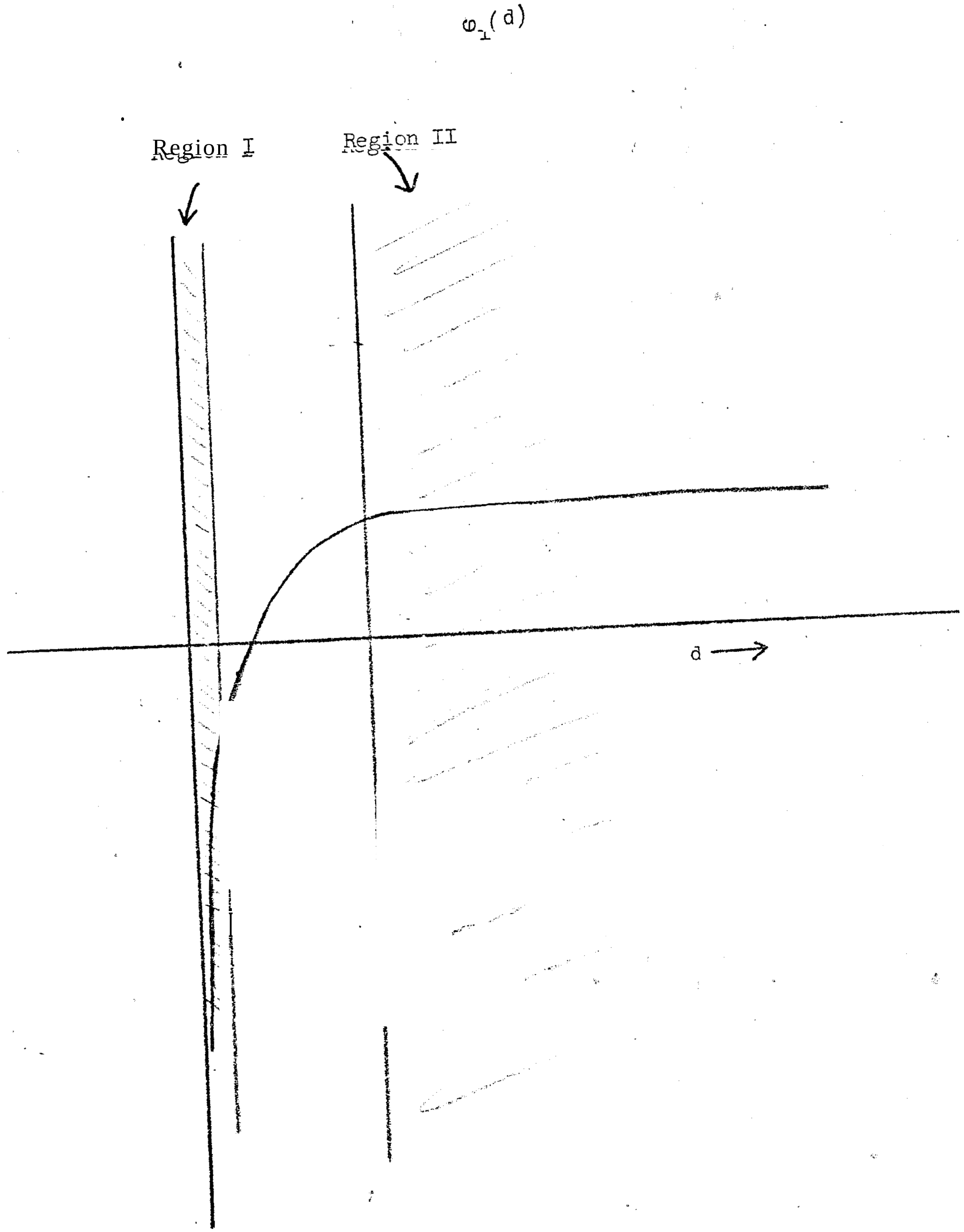


FIGURE 3.3

3.5 illustrate the behavior of Ψ_1 for two values of k_1. Even for kl small, when the nonlinear portion of Ψ_1 is more significant, Ψ_1 is very close to linear in (0,1).

Because Ψ_1 is well-behaved, Newton's method will converge to vsry rapidly. However, even further advantage can be taken of the form of Ψ_1 by noting that satisfies the relation

$$k_1 = -\hat{y} - \ln(1-\hat{y}) \quad . \tag{3.2.2}$$

Since the 'right-hand side of (3.2.2) depends only on the parameter $, and is independent of the problem dsta, the values of the function on the right-hand side can be tabulated for a set of y in (0,1). For a particular value of k_1, table lookup could then be used to determine a highly accurate estimate of $\hat{y}$. With a sufficiently large table, no iteration would be necessary to locate the solution to any desired accuracy; but since Newton's method converges rapidly, a smell table plus a single iteration is sufficient to locate y (snd $\hat{d}$) to the required accuracy.

Case 2. Three Function Values

Assuming that $8_3 > \theta_2 > \theta_1$, the three equations specifying the parameters $\hat{a}$, $\hat{b}$, and $\hat{d}$ of f_L are:

$$\text{(A)} \quad f_1 = \hat{a} + \hat{b}\theta_1 - r \ln(\hat{d}-\theta_1)$$

$$\text{(B)} \quad f_2 = \hat{a} + \hat{b}\theta_2 - r \ln(\hat{d}-\theta_2)$$

$$\text{(C)} \quad f_3 = \hat{a} + \hat{b}\theta_3 - r \ln(\hat{d}-\theta_3) .$$

There is no loss of generality in assuming that $\theta_1 = 0$.

Using equations (A) and (B) to eliminate $\hat{a}$, we obtain:

$$\frac{f_2-f_1}{\theta_2} = \hat{b} - \frac{r}{\theta_2} \ln\left(\frac{\hat{d}-\theta_2}{\hat{d}}\right)$$

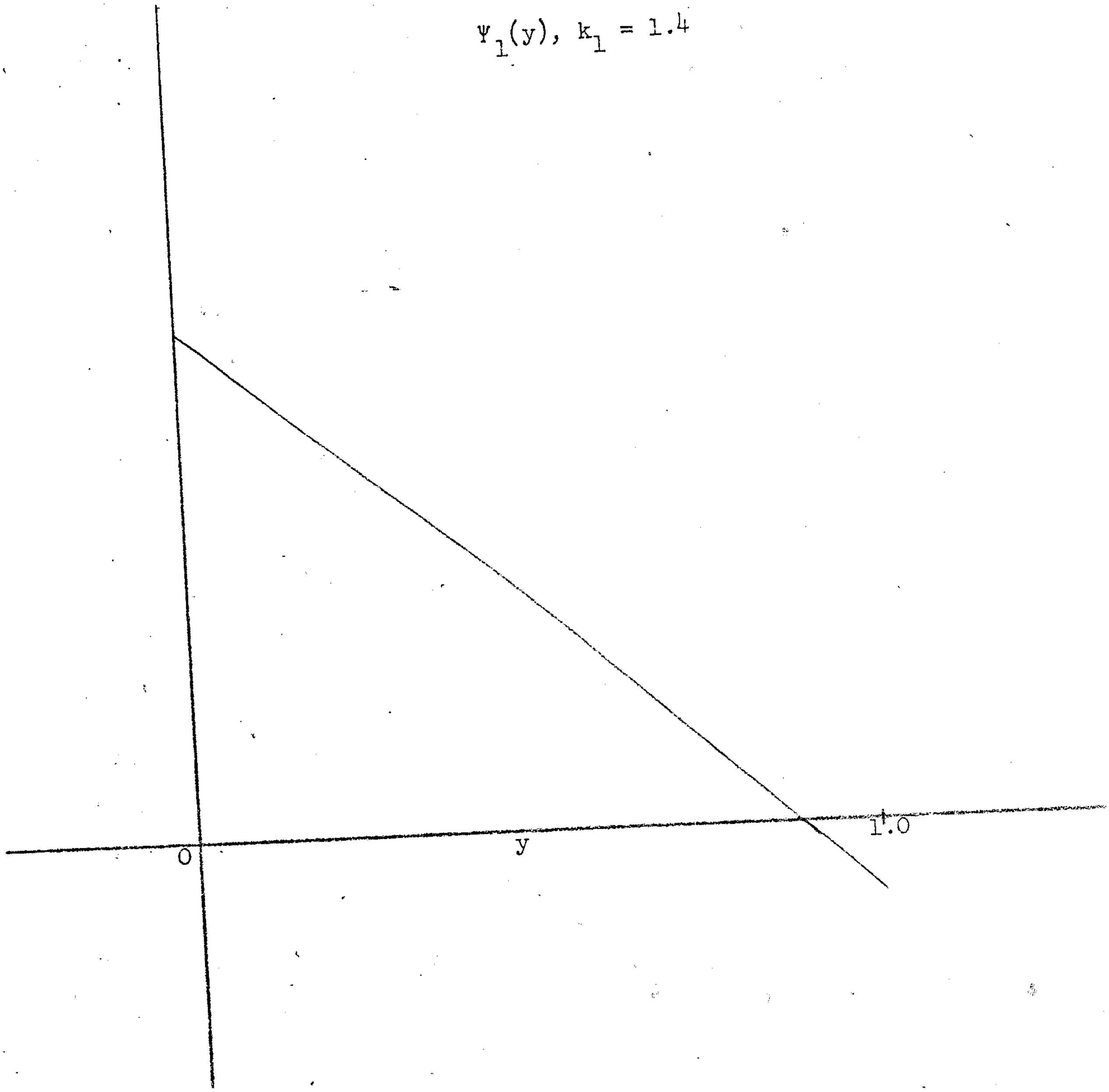


FIGURE 3.4

$\Psi_1(y)$, $k_1 = .1$

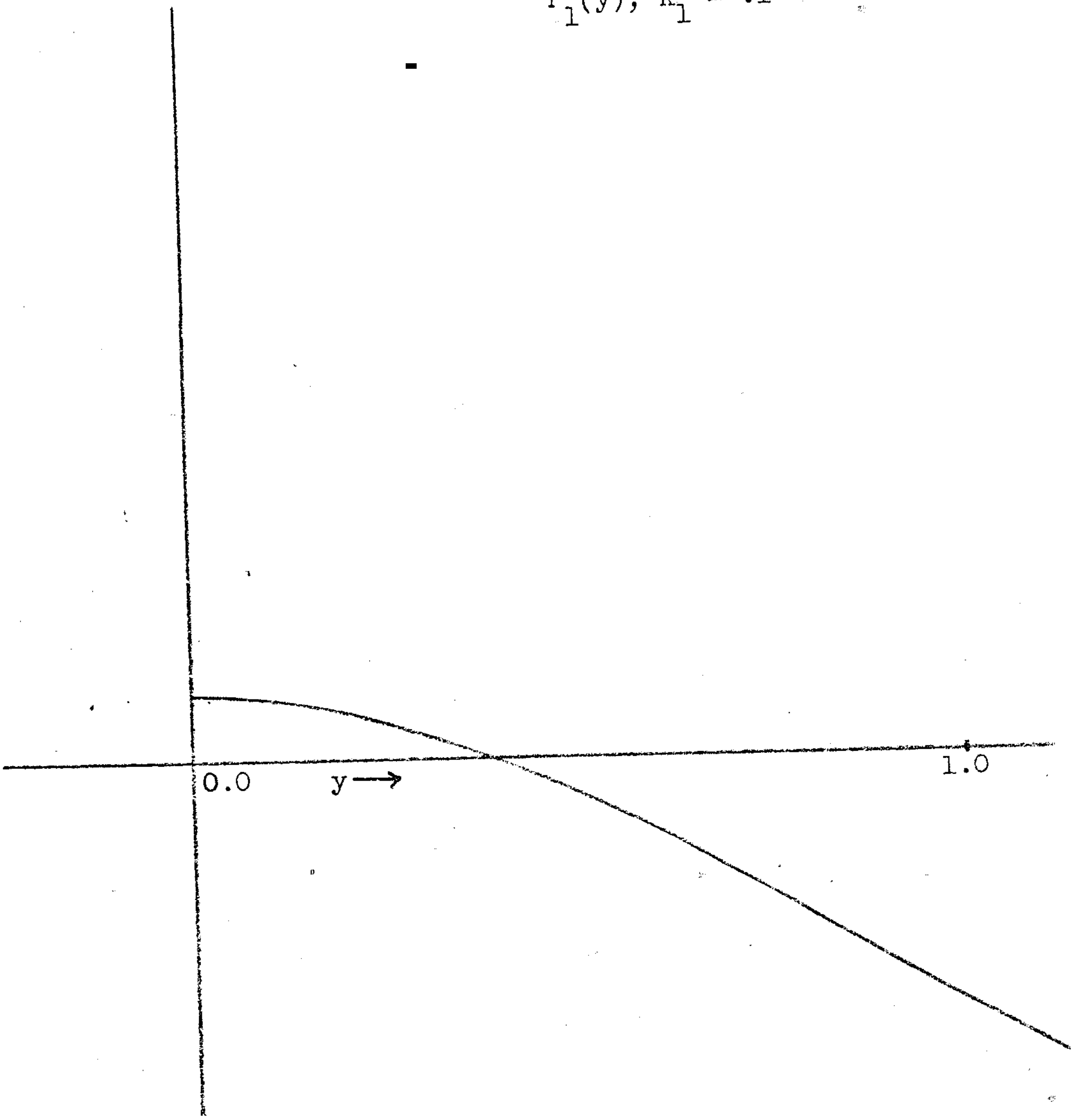


FIGURE 3.5

and ~~similarly~~; ~~using~~ (A) and (C) gives:

$$\frac{f_3 - f_1}{\theta_3} = \hat{b} - \frac{r}{\hat{}} \ln\left(\frac{\hat{d} - \theta_3}{\hat{d}}\right) .$$

These two equations can then be manipulated to eliminate $\hat{b}$; yielding:

$$\frac{1}{\theta_3} \ln\left(\frac{\hat{d} - \theta_3}{\hat{d}}\right) - \frac{1}{\theta_2} \ln\left(\frac{\hat{d} - \theta_2}{\hat{d}}\right) + \frac{1}{r}\left(\frac{f_3 - f_1}{\theta_3} - \frac{f_2 - f_1}{\theta_2}\right) = 0, \qquad (3.2.3)$$

a relationship which must be satisfied by $\hat{d}$, the location of the singularity.

Consider (3.2.3) as a nonlinear equation in terms of the function

$$\Phi_2(d) = \frac{1}{\theta_3} \ln\left(\frac{d - \theta_3}{d}\right) - \frac{1}{\theta_2} \ln\left(\frac{d - \theta_2}{d}\right) + k_2 ,$$

where $k_2 = \frac{1}{r}\left(\frac{f_3 - f_1}{\theta_3} - \frac{f_2 - f_1}{\theta_2}\right)$. The problem to be solved then becomes that of finding a solution $\hat{d}$ to satisfy $\Phi_2(d) = 0$.

Because θ_2 and θ_3 are feasible, the value of $\hat{d}$ must lie in the interval (θ_2, ∞). The function Φ_2 has the following properties:

$\Phi_2 \rightarrow -\infty$ as $d \rightarrow \theta_3 +$;

$\Phi_2 \rightarrow k_2$ as $d \rightarrow \infty$;

$\Phi_2' > 0$ for all finite d ;

$\Phi_2' \rightarrow \infty$ as $d \rightarrow \theta_3 +$;

$\Phi_2' \rightarrow 0$ as $d \rightarrow \infty$; and

$\Phi_2'' < 0$ for all finite d.

These relations imply that there is a unique zero of Φ_2 in (θ_3, ∞) if $k_2 > 0$. This requirement has a similar interpretation to Case 1, i.e., the configuration of f_1, f_2, and f_3 must be as shown in Figure 3.6, so that the line joining f3 and f1 must lie above the line joining f2 and f_1.

Because of its highly nonlinear behavior, Φ_2 is unsuited for application of the usual zero-finding techniques. The behavior of Φ_2 is illustrated in Figure 3.7.

To transform the problem of finding $\hat{d}$ to solve $\Phi_2(d) = 0$ into a computationally manageable form, we introduce the variables $\bar{d} = \frac{d}{\theta_2}$, so that $\bar{d} \gtrsim 1$, and $\bar{\theta}_3 = \frac{\theta_3}{\theta_2}$, with $\bar{\theta}3 > 1$; in essence, we consider θ_2 to be unity and scale θ_3, and d accordingly. If we multiply the equation $\Phi_2(d) = 0$ by θ_3 and substitute the variables $\bar{d}$ and $\bar{\theta}_3$,

the equation $\Phi_2(d) = 0$ by θ_3 and substitute

we obtain:

$$\ln\left(1 - \frac{\bar{\theta}_3}{\bar{d}}\right) = \bar{\theta}_3 \ln\left(1 - \frac{1}{\bar{d}}\right) - k_2\theta_3 \qquad (3.2.4)$$

$\bar{d}$

d

missible d. Taking exponentials of both sides cf (3.2.4) gives:

missible d. Taking exponentials of both sides cf (3.2.4) gives:

$$(1-y) = \left(1 - \frac{y}{-}\right)^{\bar{\theta}_3} e^{-k_2\theta_3},$$

and the value $\hat{y}$ which satisfies this relation is a zero of the function:

and the value y which satisfies this relation is a zero of the func-

$$\Psi_2(y) = 1-y-\beta\left(1 - \frac{y}{\gamma}\right)^{\gamma},$$

where

$$\gamma = \bar{\theta}3 > 1, \text{ and } \beta = e^{-k_2\theta_3} \text{ (so that } 0 < \beta < 1 \text{ for } k_2 > 0).$$

Differentiating Ψ_2 with respect to y, we obtain

$$\Psi_2' = -1 + \beta\left(1 - \frac{y}{\gamma}\right)^{\gamma-1};$$

$$\Psi_2'' = -\beta\,\frac{\gamma-1}{\gamma}\left(1 - \frac{y}{\gamma}\right)^{\gamma-2}.$$

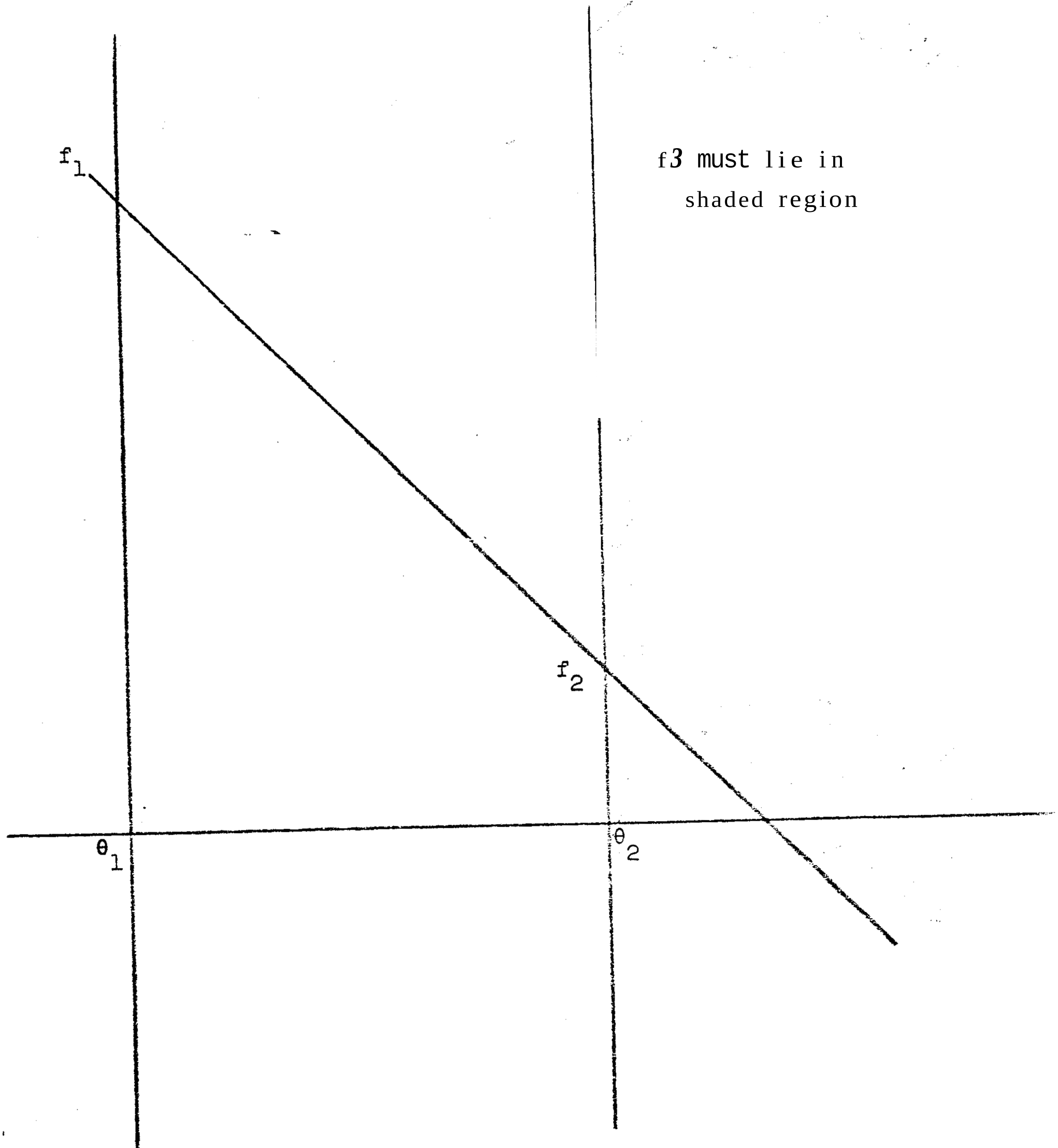


FIGURE 3.6

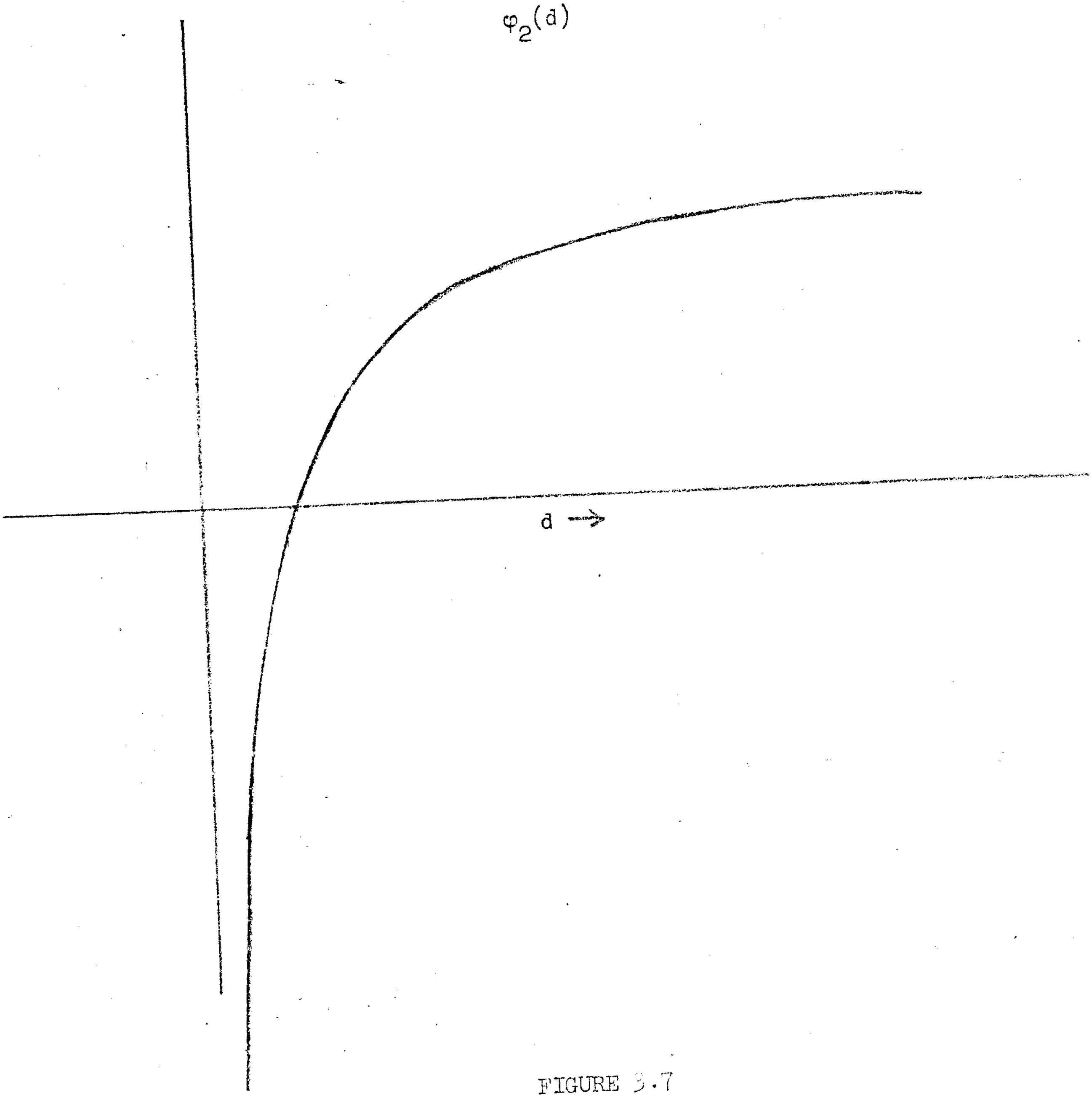


FIGURE 3.7

Y2 has the following properties:

$\Psi_2(0) = 1-\beta > 0$ (y=0 corresponds to d → ∞);

$\Psi_2(1) = -\beta\left(1 - \frac{1}{\gamma}\right)^{\gamma} < 0$ (y=1 corresponds to d → θ3+);

$Y2'(0) = -1 + \beta < 0$

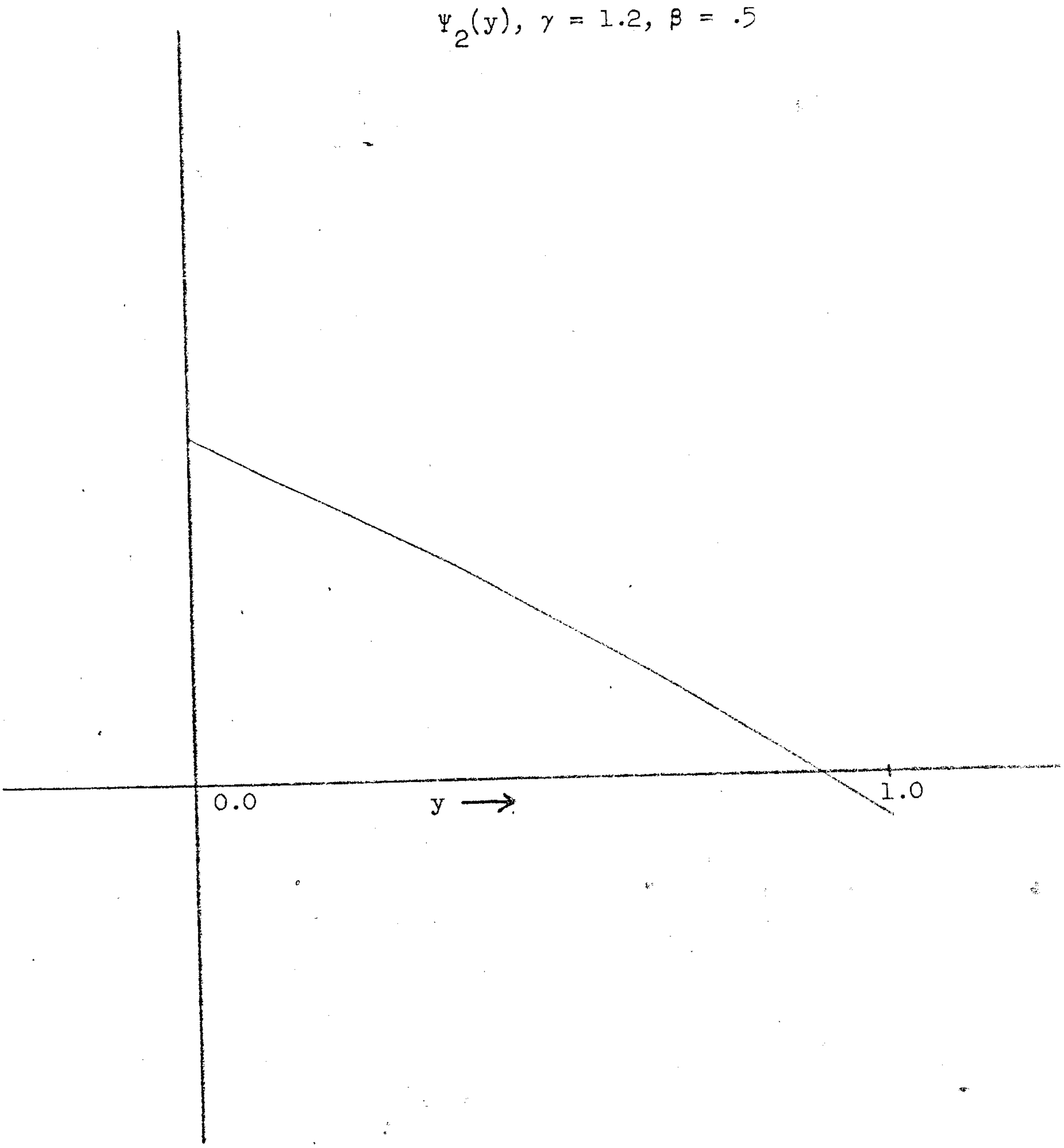
$\Psi_2'(1) = -1 + \beta\left(1 - \frac{1}{\gamma}\right)^{\gamma-1} < 0$, and

$Y2'' < 0$ for y in the interval (0,1).

These-conditions imply that Ψ_2 has a unique zero in (0,1), which could be located by Newton's method. Y2 is a well-behaved function, in contrast to Φ_2. The nonlinearity in Ψ_2 results from the expression $\left(1 - \frac{y}{\gamma}\right)^{\gamma}$, where y > 1, and is the quotient $\frac{9_3}{\theta_2}$ from the original problem. The value of y has been monitored during several runs with particular barrier functions; it never exceeded 2.0, and was usually in the range (3.01, 1.5) Even if γ were large, the expression $\left(1 - \frac{y}{\gamma}\right)^{\gamma}$ approaches e^{-y} as y approaches ∞, and hence the function Ψ_2 takes on the form of Ψ_1 in Case 1. Furthermcre, the value of $\beta(= e^{-k2\theta3})$ is usually small, so that the nonlinear component of Ψ_2 is even less significant.

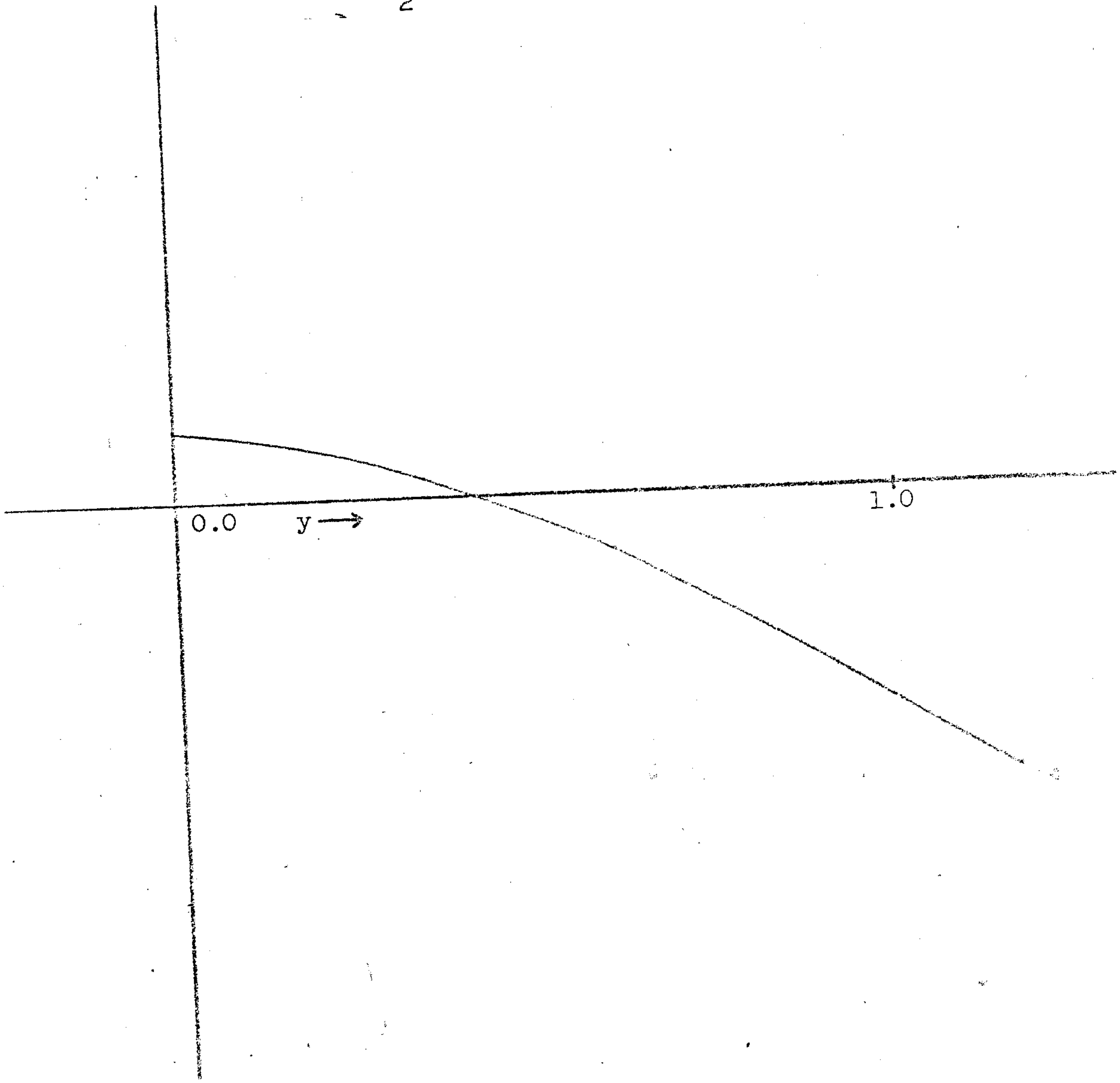
Figures 3.8 and 3.9 illustrate the behavior of Ψ_2 for two sets of parameters (β,γ). Note that even for large β and γ, the nonlinearities do not have a significant effect.

It is not computationally ccnvenient to obtain a tabulation of Ψ_2 in order to find an initial estimate of $\hat{y}$, since the influences of γ and y are not separable. However, Newton's method will. converge rapidly from a reasonable starting point (for example, the estimate from Case 1).



$\Psi_2(y)$, $\gamma = 1.2$, $\beta = .5$

FIGURE 3.8



$\Psi_2(y)$, $\gamma = 7.0$, $\beta = .9$

FIGURE 3.9

3.2.3 Quadratic Function plus Logarithmic Singularity

The special function to be fitted is of the form:

$$f_Q = \hat{a} + \hat{b}\theta + \hat{c}\theta^2 - r\, \ln(\hat{d}-\theta).$$

Differentiating f_Q with respect to θ gives:

$$f_Q' = \hat{b} + 2\hat{c}\theta + \frac{r}{\hat{d}-\theta},$$

$$f_Q'' = 2\hat{c} + \frac{r}{(\hat{d}-\theta)^2}, \text{ and}$$

$$f_Q''' = \frac{2r}{(\hat{d}-\theta)^3}.$$

Since the barrier function is decreasing at $\theta=0$, this same condition will be required of f_Q, so that $f_Q'(0) < 0$. Hence, there must exist a minimum of f_Q in the interval $(0,\hat{d})$, because $f_Q \rightarrow \infty$ as $\theta \rightarrow \hat{d}$.

A stationary point of f_Q will occur at θ^* where $f_Q'(\theta^*)=0$, so that θ^* satisfies

$$\hat{b} + 2\hat{c}\theta^* + \frac{r}{\hat{d}-\theta^*} = 0.$$

This relation leads to a quadratic equation satisfied by θ^*,

$$2\hat{c}\theta^2 + (\hat{b}-2\hat{c}\hat{d})\theta - r - \hat{b}\hat{d} = 0,$$ with two solutions which can be written in the alternative forms:

$$\theta^* = \frac{2\hat{c}\hat{d}-\hat{b} \pm \sqrt{(\hat{b}-2\hat{c}\hat{d})^2 + 8\hat{c}\,(r+\hat{b}\hat{d})}}{4\hat{c}}, \text{ or} \qquad (3.2.5)$$

$$\theta^* = \frac{2\hat{c}\hat{d}-\hat{b} \pm \sqrt{(\hat{b}+2\hat{c}\hat{d})^2 + 8r\hat{c}}}{4\hat{c}}. \qquad (3.2.6)$$

We now examine the properties of these two solutions in order to determine which will qualify as a suitable minimum, i.e., lie in the interval $(0,\hat{d})$. Since $f_Q'(0) < 0$, and $\hat{d} > 0$, the expression

$\hat{b}\hat{d} + r$ is negative. Therefore, the sign of the term $8\hat{c}(r+\hat{b}\hat{d})$ is opposite to the sign of $\hat{c}$.

Consider two cases:

Case A: $\hat{c} > 0$ (the quadratic term 4n f_Q has a positive coefficient).

The quantity $(\hat{b} - 2\hat{c}\hat{d})^2 + 8r\hat{c}$ involved in formula (3.2.6) for θ^* must be positive, so that two real roots exist. In formula (3.2.5) for θ^*, the quantity under the square root has magnitude less than $|2\hat{c}\hat{d}-\hat{b}|$ since $8\hat{c}(r+\hat{b}\hat{d})$ is negative. The expression $2\hat{c}\hat{d}-\hat{b}$ is positive since $\hat{c} > 0$, $\hat{d} > 0$, $\hat{b} < 0$, and therefore, both roots are positive. From formula (3.2.6) for θ^* we ncte that the quantity under the square root exceeds $|\hat{b} + 2\hat{c}\hat{d}|$ in magnitude.

Case A-i: $\hat{b} + 2\hat{c}\hat{d} > 0$.

The value of θ^* corresponding to the positive square root satisfies:

$$\theta^*_+ > \frac{(2\hat{c}\hat{d}-\hat{b}+\hat{b}+2\hat{c}\hat{d})}{4\hat{c}} = \hat{d}, \text{ and hence is unacceptable.}$$

The value of θ^* for the negative square root satisfies:

$$\theta^* < 2\hat{c}\hat{d}-\hat{b}-(\hat{b}+2\hat{c}\hat{d}) = \frac{-\hat{b}}{\hat{c}} .$$

Since $\hat{b} + 2\hat{c}\hat{d}$ is positive, and $\hat{c}$ is positive, it follows that $\hat{d} > -\frac{\hat{b}}{c}$, and θ^*_- is an acceptable choice for a minimum.

<u>Case A-ii:</u> $\hat{b} + 2\hat{c}\hat{d} < 0$ (i.e., $|\hat{b} + 2\hat{c}\hat{d}| = -\hat{b} - 2\hat{c}\hat{d}$).

T--- the root corresponding to the positive square root satisfies:

$$\theta^*_+ > \frac{(2\hat{c}\hat{d} - \hat{b}-\hat{b}-2\hat{c}\hat{d})}{4\hat{c}} = -\frac{\hat{b}}{\hat{c}}$$

Since $\hat{b} + 2\hat{c}\hat{d} < 0$, $\hat{d}$ must be less than $-\frac{\hat{b}}{c}$; thus, $\theta^*_+ > \hat{d}$, and is unacceptable.

The solution corresponding to the negative square root satisfies:

$$\theta^*_- < \frac{(2\hat{c}\hat{d} - \hat{b} +(\hat{b}+2\hat{c}\hat{d})}{4\hat{c}} = \hat{d},$$

and is acceptable.

Case B: $\hat{c} < 0$ (the coefficient of the quadratic term in f_Q is negative).

The quantity under the square root must be positive since $\hat{c} < 0$, $r + \widehat{bd} < 0$, and thus there are two real roots. In formula (3.2.5) for θ^*_- the quantity under rhe square root exceeds $2\hat{c}\hat{d}-\hat{b}$ in magnitude, and hence there must be one positive and one negative root. Because of the sign reversal caused by division by c, the positive square root corresponds to a value $\theta^* < 0$, and can be eliminated from consideration. The only positive θ^* can be written as:

$$\theta^* = \frac{\hat{b} - 2\hat{c}\hat{d} + \sqrt{(\hat{b}+2\hat{c}\hat{d})^2 + 8r\hat{c}}}{4|c|} .$$

The quantity under the square root has magnitude less than $|\hat{b} + 2\hat{c}\hat{d}|$, so that θ^* satisfies:

$$\theta^* < \frac{\hat{b} - 2\hat{c}\hat{d} - \hat{b}-2\hat{c}\hat{d}}{4|\hat{c}|} = - \frac{4\hat{c}\hat{d}}{4|\hat{c}|} = \hat{d}.$$

For Cases A and B, then, assuming that $f_Q'(0) < 0$, the solution θ^* corresponding to the negative square root in formula (3.2.5) or (3.2.6) will be taken as the minimum of f_Q.

In order to specify the function f_Q, four independent pieces of information are required about the behavior of the function to be fitted along the search direction; f_Q will therefore be used in circumstances where a cubic fit would normally be carried out to minimize a general function. Figure 3.10 illustrates a r = .001

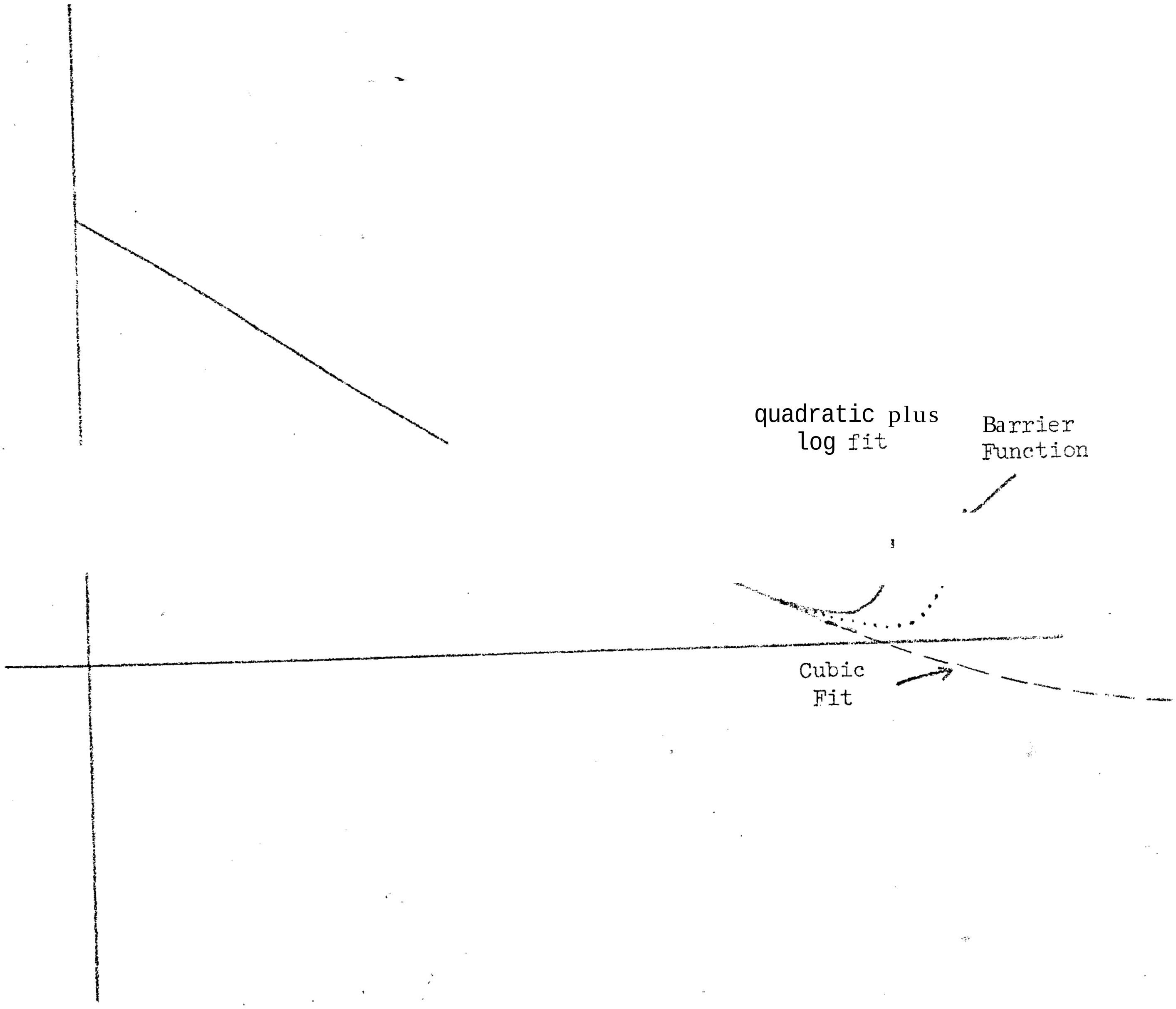


FIGURE 3.10

barrier function and its approximations by the function f_Q and a cubic, both using the same data. The more accurate moaeling of the barrier function by the special function is quite noticeable.

The special function fQ will be fitted with the same set of data used by a typical cubic line search, namely, two function values and the corresponding two gradients. As in the previous discussion, f_i will denote the function value at θ_i, and g_i is the corresponding gradient value.

The four unknown parameters of fQ - $\hat{a}$, $\hat{b}$, $\hat{c}$, and $\hat{d}$ -- must be solved for in terms of the known values. Assuming thst $\theta_2 > \theta_1$, the equations specifying fQ are:

$$\text{(A)} \quad f_1 = \hat{a} + \hat{b}\theta_1 + \hat{c}\theta_1^2 - r \ln \text{(a-e,)}$$

$$\text{(B)} \quad g_1 = \hat{b} + 2\hat{c}\theta_1 + \frac{r}{(\hat{d}-\theta_1)}$$

$$\text{(C)} \quad f_2 = \hat{a} + \hat{b}\theta_2 + \hat{c}\theta_2^2 - r \ln(\hat{d}-\theta_2)$$

$$\text{(D)} \quad g_2 = \hat{b} + 2\hat{c}\theta_2 + \frac{r}{(\hat{d}=\theta_2)}$$

There is no loss of generality in assuming that $\theta_1 = 0$, so that from (A) and (B) we obtain expressions for the coefficients $\hat{a}$ and $\hat{b}$ in terms of $\hat{d}$:

$$\hat{a} = f_1 + r \ln \hat{d}$$

$$\hat{b} = g_1 - \frac{r}{\hat{d}}$$

Substituting for $\hat{b}$ in equation (D), we obtain an expressicn for the parameter $\hat{c}$ in terms of $\hat{d}$:

$$\hat{c} = \frac{1}{2\theta_2} \left(g_2 - g_1 + \frac{r}{\hat{d}} - \frac{r}{\hat{d}-\theta}\right).$$

These three expressions for $\hat{a}$, $\hat{b}$, and $\hat{c}$ can then be substituted into equation (C), to obtain the following:

$$\ln(\hat{d}-\theta_2) - \ln\hat{d} + \frac{\theta_2}{2}\left(\frac{1}{\hat{d}} + \frac{1}{\hat{d}-\theta_-}\right) = \frac{1}{r}\left(\frac{\theta_2}{2}(g_1+ \quad) \quad (f \quad f)\right) \qquad 7)$$

If a value $\hat{d}$, the location of the singularity, can be found which satisfies (3.2.7), the values $\hat{a}$, $\hat{b}$, $\hat{c}$, and θ^* can then be computed from the previously derived relationships.

Consider (3.2.7) as a nonlinear equation in terms of the function

$$\Phi(d) = \ln\left(\frac{d-\theta_2}{d}\right) + \frac{\theta_2}{2}\left(\frac{1}{d} + \frac{1}{d-\theta}\right) - k$$

where $k_3 = \frac{1}{r}\left(\frac{\theta_2}{2}(g1+g2) - (f_2-f_1)\right)$. The problem to be solved then becomes that of finding a solution $\hat{d}$ to satisfy

$$\Phi_3(d) = 0. \qquad (3.28)$$

If we introduce the variable $z = 1-\frac{\theta_2}{d} = \frac{(d-\theta_2)}{d}$, where $0 < z < 1$, we can then write the equation $\Phi_3(d) = 0$ in terms of the variable z as:

$$\ln z + \frac{1}{\alpha}\left(\frac{1}{z} - z\right) = k_\alpha \qquad (3.2.9)$$

The nonlinear equation (3.2.9) could be solved for a suitable $\hat{z}$, but the function represented is extremely ill-behaved. As $z \to 0$, the logarithm term is approaching $(-\infty)$, while the reciprocal term is simultaneously approaching $(+\infty)$; it is evident that (3.2.9) is quite unsuitable for purposes of computation.

However, the relationship (3.2.9) can be transformed into an equivalent form that is computationally reasonable. A further change of variable is made:

$$v = \ln z,$$

so that $z = e^v$ and $\frac{1}{z} = e^{-v}$; note that v will be nonpositive since $0 < z < 1$.

When written in terms of v, the relation (3.2.9) becomes:

$$v + \frac{1}{2}\left(e^{-v} - e^{v}\right) = k_3 \ .$$

Since $\sinh(v) = \frac{1}{2}\left(e^{v} - e^{-v}\right)$, the final result is:

$$v - \sinh(v) = k3 .$$

The value $\hat{v}$ that satisfies this relation is a zero of the function Ψ_3, where

$$\Psi_3(v) = k3 + \sinh(v) - v.$$

Differentiating $Y3$ with respect to v, we obtain:

$$\Psi_3' = \cosh(v) - 1$$

$$\Psi_3'' = \sinh(v),$$

so that the Ψ_3 has the following properties:

$\psi3(0) = k3$ (v=0 corresponds to $d \to \infty$);

$$\lim_{v \to -\infty} \Psi_3(v) = -\infty;$$

$$\Psi_3' > 0 \text{ for } v < 0;$$

$$\Psi_3'(0) = 0;$$

$$\lim_{v \to -\infty} \Psi_3'(v) = \infty;$$

$$\Psi_3'' < 0 \text{ for } v < 0.$$

These conditions imply that Ψ_3 has a unique zero in $(-\infty, 0)$ **if** $k_3 > 0$. This requirement means that the average of the gradients at θ_1 and θ_2 must exceed the slope of the straight line joining f_1 and f_2. If the function to be approximated were quadratic, the average of the slopes at θ_1 and $^{\theta}2$ would exactly equal the slope of the line joining f1 and f_2. The condition $k_3 > 0$ thus implies that the function to be approximated is rising more rapidly than a quadratic.

Figure 3.11 illustrates the behavior of Ψ_3.

Although the function Y3 is unbounded below as $v \to -\infty$ (i.e., when $z \to 0$, or $d \to \theta_2 +$), this property does not cause any computational difficulties in the current context. The unbounded behavior of Y_3 occurs when the estimated value of the singularity is very close to θ_2; if a tolerance, say ϵ, is specified such that any estimate of $\hat{d}$ is required to satisfy $\hat{d} \geq \frac{\theta_2}{1-\epsilon}$, then the variable z is bounded below by ϵ, and the variable v is bounded below by $-M$, $M > 0$, where $M = -\ln(\epsilon)$. If values of v are restricted to the range $(-M,0)$, the region where Y3 is unbounded is eliminated. If k_3 is very large, it is possible that the value $\Psi_3(-M)$ will not *be* negative for the particular value of M chosen, and hence no zero of Y_3 will exist in $(-M,0)$ Under these circumstances, we simply accept $\hat{v} = -M$ as the solution, so that $\hat{d} = \frac{\theta_n}{1-\epsilon}$.

We can easily solve the equation $Y3(v) = 0$ with Newton's method, considering the following properties of $Y3$. Since $\Psi3'' < 0$ throughout the interval of interest, if the starting point is chosen so that $\Psi3 < 0$, the Newton iterates will undershoot the solution, and cannot diverge. Furthermore, the condition $\Psi3(v) = 0$ can be written as:

$$k_3 = v - \sinh(v),$$

and the expression on the right-hand side is independent of the problem data. Hence, the function $v - \sinh(v)$ can be tabulated for v in the range $(-M,0)$, and by table lookup using the value $k3$, a highly

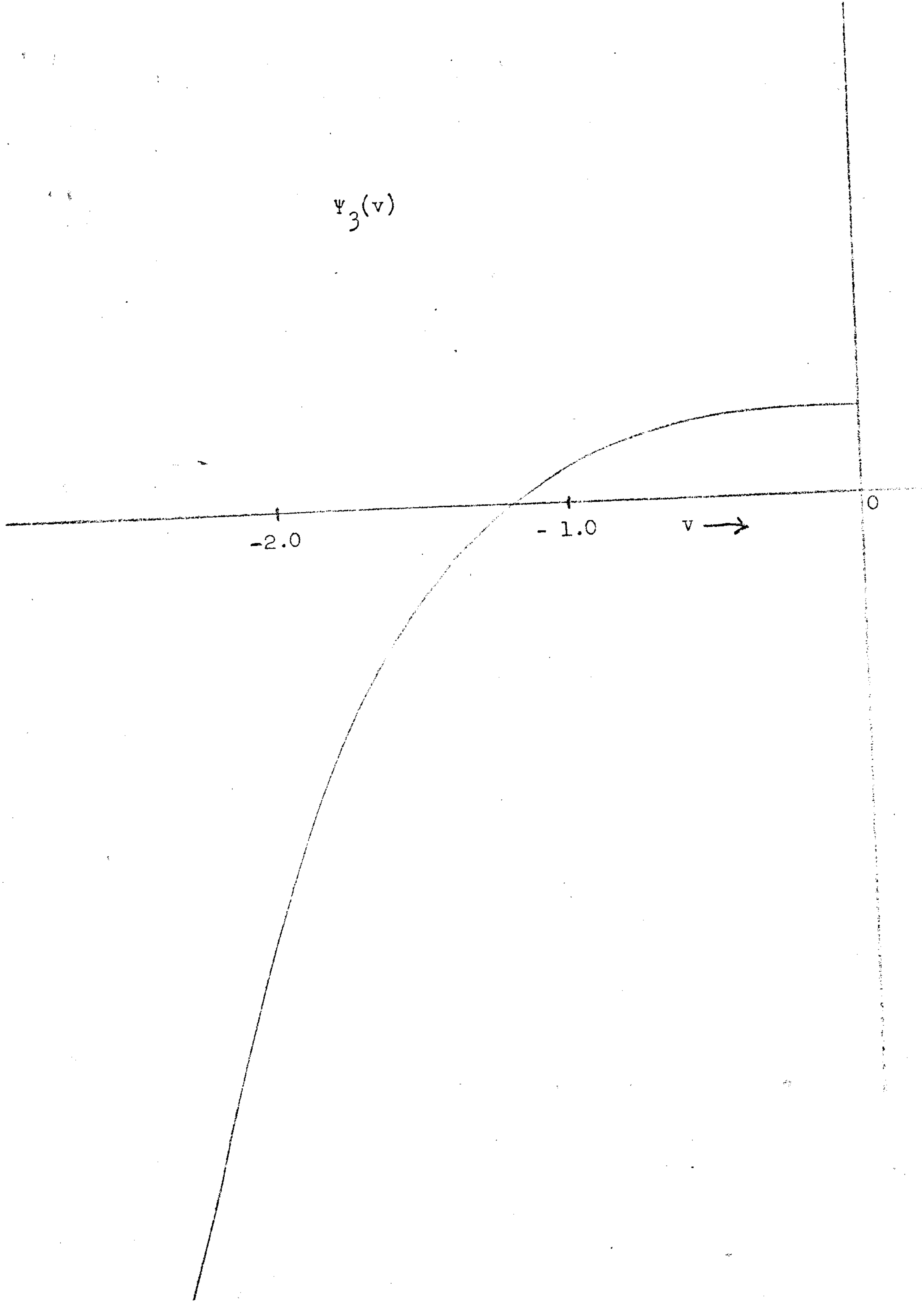


FIGURE 3.11

accurate initial estimate of v, with $\Psi_3 < 0$, can be obtained.

The Newton procedure converges extremely rapidly except when the value of v is close to zero, because $\Psi_3'(0) = 0$. This situation is quite unlikely because the estimated singularity would then be much larger than θ_2. For completeness, however, we note that the problem can be solved successfully even for very small v. The function $\Psi_3(v)$ can be written:

$$\begin{aligned} y3(v) &= k3 + \sinh (v) - v \\ &= k_3 + \frac{1}{2}(e^v - e^{-v}) - v \\ &= k_3 + \frac{1}{2}\left(1 + v + \frac{v^2}{2} + \frac{v^3}{6} + \dots - \left(1 - v + \frac{v^2}{2} - \frac{v^3}{6} + \dots\right)\right) - v \\ &= k_3 + \frac{1}{2}\left(2v + \frac{2v^3}{6} + \frac{2v^5}{120} + \dots\right) - v \\ &= k_3 + \frac{v^3}{6} + O(v^5). \end{aligned}$$

For small v, the equation $\Psi_3(v) = 0$ thus essentially becomes the condition

$$k_3 + \frac{v^3}{6} = 0,$$

with explicit solution $v^* = (-6k_3)^{\frac{1}{3}}$. Because we are ignoring negative higher-order terms, this value v^* will be to the left of the correct $\hat{v}$, and Newton's method cannot diverge. However, the estimate v^* is so accurate that no iteration at all is necessary to obtain an acceptable solution.

3.3 Implementation

The safeguarded linear searches based on quadratic or cubic interpolation (cf. Gill and Murray, 1974a) 'rave been modified for use with the logarithmic barrier function by allowing interpolation with the special functions described. Several rather complicated modifications are required in order to create an efficient algorithm. If no constraint is decreasing along the current search direction, or if nc constraint approaches zero until sufficiently far beyond the starting point, then the singularity introduced to preserve feasibility will have no significant effect on the location of the minimum, and the usual linear search procedure should be followed. There is no computationally reaeonable way to determine a priori whether these conditions exist because the constraints and objective function may be. highly nonlinear, and the effort expended to compare the location of the nearest constraint zero with the prediction-of the barrier function's minimum might be better used directly to minimize the barrier function. The procedure tc be described seems to be a satisfactory compromise between excessive safeguards and unwarranted assumptions of linearity or smoothness .

3.3.1 Initial Step

The choice of the first step along the search direction at which the function is to be evaluated is affected by rhe possibility that a constraint may become nonpositive if the usual chcice of step for the algorithm is taken. For example, with a Newton-type method, the initial step taken along the search direction is unity; for a quasi-Newton method, there is normally a procedure asscciated with the method for choosing the initial step. Let α_u denote the initial step that would be taken for a particular unconstrained algcrithm if used to minimize the barrier function along the given direction. If a constraint might become zero at $\hat{\alpha} < \alpha_u$, clearly a shorter step than α_u should be taken. One possible method for determining the initial step is to find a highly accurate estimate of the step to the nearest zero of a constraint, say $\hat{\alpha}$, and test whether $\hat{\alpha} < \alpha_u$. A subroutine is available that will, with high reliability, locate the zero of the nearest constraint by use of a combination of safeguarded zero-finding techniques . However, locating $\hat{\alpha}$ generally requires several constraint evaluations, and it may turn out that $\hat{\alpha}$ exceeds α_u or is very close to cyU, so that these evaluations were essentially redundant. One might think of using the zero-finding technique until the zero has been shown conclusively to lie beyond $\hat{\alpha}$, but this approach invclves quite complicated housekeeping, and, more significantly, may still require constraint calculations that do not advance the computation.

With the "compromise" algorithm, the initial step $\alpha^{(0)}$ is computed as follows:

(1) Compute cy_u, the step normally taken by the unconstrained method;

(2) Compute the gradient of each constrain", along p, i.e., $a_i^T p$. For all i such that this gradient is negative, i.e., the ith constraint is locally decreasing along p, compute $\alpha_i = -c_i/a_i^T p$, the predicted Newton step to the zero of c_i. Find $\bar{\alpha} = \min(\alpha_i)$, and let $\bar{i}$ be the index for which $\bar{\alpha} = \alpha_{\bar{i}}$. In other words, $\bar{\alpha}$ is the smallest positive first-order step to a constraint zero. If no constraint is decreasing along p, set the initial step $cy^{(0)}$ to α_u, and skip the remaining logic.

(3) Estimate the step to the minimum of the barrier function, as follows. The projected gradient of the barrier function will be zero at the minimum along the line, and we assume that only the influence of the constraint $\bar{i}$ will be significant in the location of %heminimum.

This assumption is based on the idea that for small r, only the constraint whose zero is closest to the starting point will affect the barrier function by altering the behavior of the objective function. A crude estimate for the minimum of the barrier function, α_b, may be obtained by assuming that the gradients of the objective function and $\underline{\bar{i}}$th constraint will remain fixed locally. Thus, we obtain the following relation involving α_b:

$$p^T(g(x+\alpha_b p) - \sum_{i=1}^{t} \frac{r}{c_i(x+}$$

Ignoring all but the $\underline{\bar{i}}$th constraint and second-order terms gives:

$$p^T g - \frac{r a_{\bar{i}}^T p}{c_{\bar{i}} + \alpha_b a_{\bar{i}}^T p} = 0 \;;$$

$$\alpha_b a_{\bar{i}}^T p = \frac{r a_{\bar{i}}^T p}{T} - c_{\bar{i}} \;;$$

$$\alpha_b = \frac{r}{p^T g} - \frac{c_{\bar{i}}}{a_{\bar{i}}^T p} \;;$$

and since $\bar{\alpha} = \frac{-c_{\bar{i}}}{T}$, $\alpha_b = \bar{\alpha} + \frac{r}{T}$.

If $p^T g > 0$, i.e., p is not a descent direction for the objective function, then the assumption that only one constraint will have an effect, confined to the neighborhood of the singularity, is not justified, and we set $\alpha_b = \gamma\bar{\alpha}$ where $0 < \gamma < 1$.

(4) If α_b exceeds α_u let $\alpha^{(0)}$ be α_u; otherwise, $\alpha^{(0)} = \alpha_b$.

In this way, the initial step can be chosen based on the estimated decrease of tlie constraints if it seems that their effect will be significant in the location of the minimum of the barrier function.

3.3.3 Feasibility Check

For each step to be taken, the set of ccnstraints is evaluated in order to assure that feasibility is never violated.

If any constraint, say the jth, is non-positive at $x + \alpha^{(k)}p$, the secant step to the predicted zero cf that constraint is computed, and the next estimate of the location of the minimum is computed as described above in Step (3), where the secant step $\bar{\alpha}_j$, given by:

$$\bar{\alpha}_j = \frac{-a^{(k)} c_j(x)}{c_j(x+\alpha^{(k)}p) - c_j(x)},$$

is used as $\bar{\alpha}$. This procedure is subject to the safeguarding requirement that the constraints not be evaluated at points that are *too* close together (see Gill and Murray, 1974a; Brent, 1973, for a discussion of this aspect of safeguarded linear searches)

3.3.3 Normal Iteration of Modified Linear Search

The special functions are fitted during the iteration if a fiag has been set to 'true'. The flag is set: (1) when the *initial* $\alpha^{(0)}$ was α_b, implying that the influence of some constraint is predicted to be significant in locating the minimum; and (2) when any negative constraint value is encountered during execution of the linear search, since it has then been demonstrated that the current iterate is in a region of influence of the singularity.

The special functions are fitted iteratively, using the same criteria for replacing points as the usual linear searches, until the particular convergence criteria are satisfied. There are a few subtle difficulties in that for small r, it may be difficult to locate the minimum because the distance from the singularity to the minimum may be less than the spacing required for constrain; evaluations. However, a careful regulaticn of the tolerances involved, so that impossible accuracy is not sought, will assure that the process will work as desired.

3.3.4 Comparison with Usual Linear Searches

In order to determine whether the special linear searches are worthwhile, numerical experiments were carried out for several barrier functions, with varying values of r, the barrier parameter, and η, the linear search convergence parameter. For the cubic case where gradients are evaluated at every point, the linear search usually terminates when

$$|g(x + \alpha p)| < \eta \, |g(x)| \; ;$$

in the quadratic case, the linear search is usually terminated when the minimum is known to be bracketed in the interval $[a,b]$ and

$$\left| \frac{F(x+\alpha p) - F(x+ap)}{\alpha - a} \right| < \eta \, |g(x)| ,$$

i.e., when the linearized approximation to the gradient at $x + \alpha p$ satisfies the same test as $g(x + \alpha p)$ in the cubic case. There are other occasions when the normal linear search procedure will terminate, involving sufficient smallness of the interval of uncertainty, closeness to the maximum permitted step, etc.

Numerous runs (about 40) were made. For both the special functions and the usual polynomials, the same initial step was taken, and the same procedure was followed for determining the next point if a constraint became negative during the line search iteraticn. Hence, the only difference was in the use of the minima of special functions, rather than of cubic or quadratic polynomials, . to yield the next pcint at which the function and constraints are to be evaluated as the linear search proceeds. In every case, use of $he special function reduced the number of constraint evaluations; the reduction became progressively more significant as the value of r was reduced. The reduction in the number of function and constraint evaluations ranged from 7% to 24%, so that there was clear improvement with the special functions.

The extra work required in the linear search to fit these special functions is small. Some of the housekeeping (checking lor feasibility, etc.) must be carried out with barrier functions regardless of whether special functions are used or not. The formulations presented here allow calculation of the minimum of the fitted functions with the same information required to fit the usual polynomials. The singularity must be located through an iteration, but because of the special form of the iteration functions, we are able to obtain a highly accurate starting guess; in fact, in two of the three cases, the solution cculd be obtained by table lookup for a sufficiently large table. The iteration functions are we! l-behaved, and Newton's method will usually converge to the desired accuracy within two iterations. Each iteration to locate the singularity requires evaluation of a transcendental function, but the subsequent reduction in the number of function and constraint evaluations required to locate a satisfactory approximation to the minimum of the barrier function seems ample justification for use of the special linear searches designed to minimize barrier functions.

CHAPTER 4. Methods Based on the Lagrangian Function

CHAPTER 4

METHODS BASED ON THE LAGRANGIAN FUNCTION

4.1 Introduction

The motivation for the algorithms to be discussed in this chapter is to improve over the performance of penalty arid barrier function methods by constructing a set of sub-problems that neither contain inevitably increasing ill-conditioning, nor assure convergence only in the limit. The problem to be considered first is the equality constrained problem:

P1: minimize $F(x)$

subject to $c_i(x) = 0$, $i=1,2,\ldots,m$.

The sufficiency conditions given in Section 1.2 for a constrained minimum will be assumed to hold at x^*, since the methods to be described are all critically dependent on those properties; in addition, $A(x^*)$ will be assumed to have full rank. Under these conditions, there exists at x^* a vector λ^* of Lagrange multipliers such that:

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

The point x^* is therefore a stationary point of the Lagrangian function $L(x) = F - \lambda^{*T}c$, but is not in general a local minimum. However, the sufficiency conditions guarantee that the Hessian matrix of $L(x)$ at X^*, given by $v^2L(x) \equiv W = G - \sum_{i=1}^{m} \lambda_i^* G_i$, is positive definite restricted to the null space of $A(x^*)$; equivalently, Z^TWZ is positive definite, where the matrix Z is as defined in Chapter 1, a matrix whose columns form an orthogonal basis for the null space of A. Consequently, the Lagrangian function can display negative curvature at x^* only along directions in the rang of

$A(x^*)$, and x^* is a local minimum of $L(x)$ restricted to the null space.

The algorithms to be described in this chapter are based on ettempts to use these properties of the Lagrangian functicn to construct a sub-problem of which x^* is the solution. Two approaches to devising the sub-problem will be considered:

1) Unconstrained Minimization

A function is constructed of which x^* is an unconstrained local minimum, so that an algorithm for unconstrained minimization can be applied directly. A suitable function can be generated by augmenting the Lagrangian function through addition of a term that retains the stationary properties of x^*, but alters the Hessian in the requisite subspace to make it positive definite at x^*.

2) Linearly Constrained Minimization

A linearly constrained sub-problem is developed of which x^* is the solution. The function to be minimized is selected so that x^* will retain its stationary properties; the linear constraints are constructed so that the subspace in which the minimization takes place is restricted to that in which the Hessian at x^* is in general positive definite.

Any algorithm based on the Lagrangian function includes the inherent difficulty that the Lagrange multipliers at x^* are in general unknown. All the algorithms to be described, whether they generate an unconstrained or linearly constrained sub-problem, necessarily involve some method for estimating the Lagrange multipliers. However, the primary distinctions to be noted will be based on other aspects of the methods, since the estimation of the multipliers can usually be considered a separable

4.2 Augmented Lagrangian Function

There are various possibilities for constructing a function of which x^* is an unconstrained local minimum, but only one such function will be considered here. The augmented Lagrangian function is defined as:

$$L_A(x,\lambda ,p) = F - \lambda^T c + \frac{\rho}{2} c^T Dc, \qquad (4.2.1)$$

where ρ is a positive penalty parameter, A is an estimate of the vector of Lagrange multipliers, and D is a positive definite symmetric matrix. The quantities p, A, and D do not depend on x.

There are numerous forms for this type of augmented Lagrangian function. For example, the matrix D may be assumed to be diagonal, and the parameter ρ can be absorbed into the elements of D, so that $c^T Dc$ becomes a weighted quadratic penalty term; or the matrix D may be the identity matrix, corresponding to an unweighted quadratic penalty. The details of any particular algorithm will depend on the choice of D, but the purpose of this term is the same in all cases - to add positive curvature in the range of $A(x^*)$.

The augmented Lagrangian function (4.2.1) has the property that x^* could be found by a single unconstrained minimization if the optimal multiplier vector A^* and a suitable ρ were known.

Theorem 7: If the sufficiency conditions given in Section 1.2 hold at x^* and $A(x^*)$ has full rank, then there exists $\hat{\rho} > 0$ such that x^* is an unconstrained local minimum of $L_A(x,\lambda^*,\rho)$ for $P > \hat{\rho}$.

Proof: Differentiating L_A with respect to x gives:

$$\nabla L_A(x,\lambda^*,\rho) = g - A\lambda^* + \rho ADc. \qquad (4.2.2)$$

Since the sufficiency conditions hold at x^*, $g - A\lambda^* = 0$, and $c = 0$, so that $\nabla L_A(x^*,\lambda^*,\rho) = 0$, and x^* is a stationary point of $L_A(x,\lambda^*,\rho)$, a necessary condition for x^* to be a local minimum.

The Hessian of $L_A(x,\lambda^*,\rho)$ is given by:

$$\nabla^2 L_A(x,\lambda^*,\rho) = G - \sum_{i=1}^{m} \lambda_i^* G_i + \rho \sum_{i=1}^{m} v_i G_i + \rho ADA^T,$$

where. $v_i = d_i^T c$ for d_i' the ith column of D. At x^*, $c = 0$, and thus the Hessian at x^* reduces to:

$$\nabla^2 L_A(x^*,\lambda^*,\rho) = W + \rho ADA^T.$$

If there exists $p > 0$ such tnat $\nabla^2 L_A(x^*,\lambda^*,\rho)$ is positive definite, the sufficient conditions will be satisfied for x^* to be an unconstrained local minimum.

Let y be a unit vector, which can be written as a linear combination of the columns of Q_1 and Z, where the columns of Q_1 form an orthogonal basis for the space spanned by the columns of A (so that A^TQ_1 is nonsingular), and the columns of Z form an orthogonal basis for the null space cf A. If $y = Q_1u + Zv$, the product $y^T(W + \rho ADA^T)y$ can be written as follows:

$$v^TZ^TWZv + 2v^TZ^TWQ_1u + u^TQ_1^TWQ_1u + \rho u^TQ_1^TADA^TQ_1u.$$

To verify that the matrix is positive definite for suitably chosen ρ, this quantity must be bounded below.

For a general matrix C, $x^TCy \geq -||x||\ ||C||\ ||y||$.

Because the columns of Q_1 and Z are orthogonal, $||Q_1u|| = ||u||$, and $||Zv|| = ||v||$. Since Z^TWZ is positive definite at x^* by the sufficiency conditions, it follows that:

$$v^TZ^TWZv > \alpha||v||^2$$

for some $\alpha > 0$.

The matrix D is positive definite, and Q_1^TA is nonsingular; thus $Q_1^TADA^TQ_1$ is also positive definite, satisfying:

$$u^T Q_1^T A D A^T Q_1 u \geq \beta \| u \|^2 ,$$

for some $\beta > 0$. Let γ denote $\| W \|$.

Therefore, using all these bounds, we obtain:

$$y^T (W + \rho A D A^T) y \geq \alpha \| v \|^2 - 2\gamma \| v \| \, \| u \| - \gamma \| u \|^2 + \rho\beta \| u \|^2 .$$

If $\| u \| = 0$, the expression on the right-hand side is simply $\alpha \| v \|^2$, which is known to be positive (this case corresponds to a vector y in the null space of A). Otherwise, if $\| u \| \neq 0$, the right-hand side will be positive if:

$$\rho > \frac{1}{\beta} (- \alpha k^2 + 2\gamma k + \gamma), \qquad (4.2.3)$$

where $k = \dfrac{\| v \|}{\| u \|}$. The expression on the right-hand side of (4.2.3) is maximized when $k = \frac{\gamma}{\alpha}$, and has the value $\frac{1}{\beta} (\frac{\gamma^2}{\alpha} + \gamma)$. Therefore, if ρ is chosen so that $\rho > \frac{1}{\beta} (\frac{\gamma^2}{\alpha} + \gamma)$, the matrix $W + \rho A D A^T$ will be positive definite, and x^* will be a local unconstrained minimum of $L_A(x, \lambda^*, \rho)$.

Figure 4.1 illustrates that x^* is a local unconstrained minimum of $L_A(x, \lambda^*, \rho)$, but only a stationary point of $L_A(x, \lambda^*, 0)$, the ordinary Lagrangian function.

In practice, of course, λ^* is generally unknown. Furthermore, the inequality that must be satisfied by ρ is based on quantities evaluated at x^*, and hence a suitable choice for ρ is also unknown. The methods for solving P1, based on this augmented Lagrangian function, must therefore include procedures for estimating λ^* and for choosing an appropriate ρ.

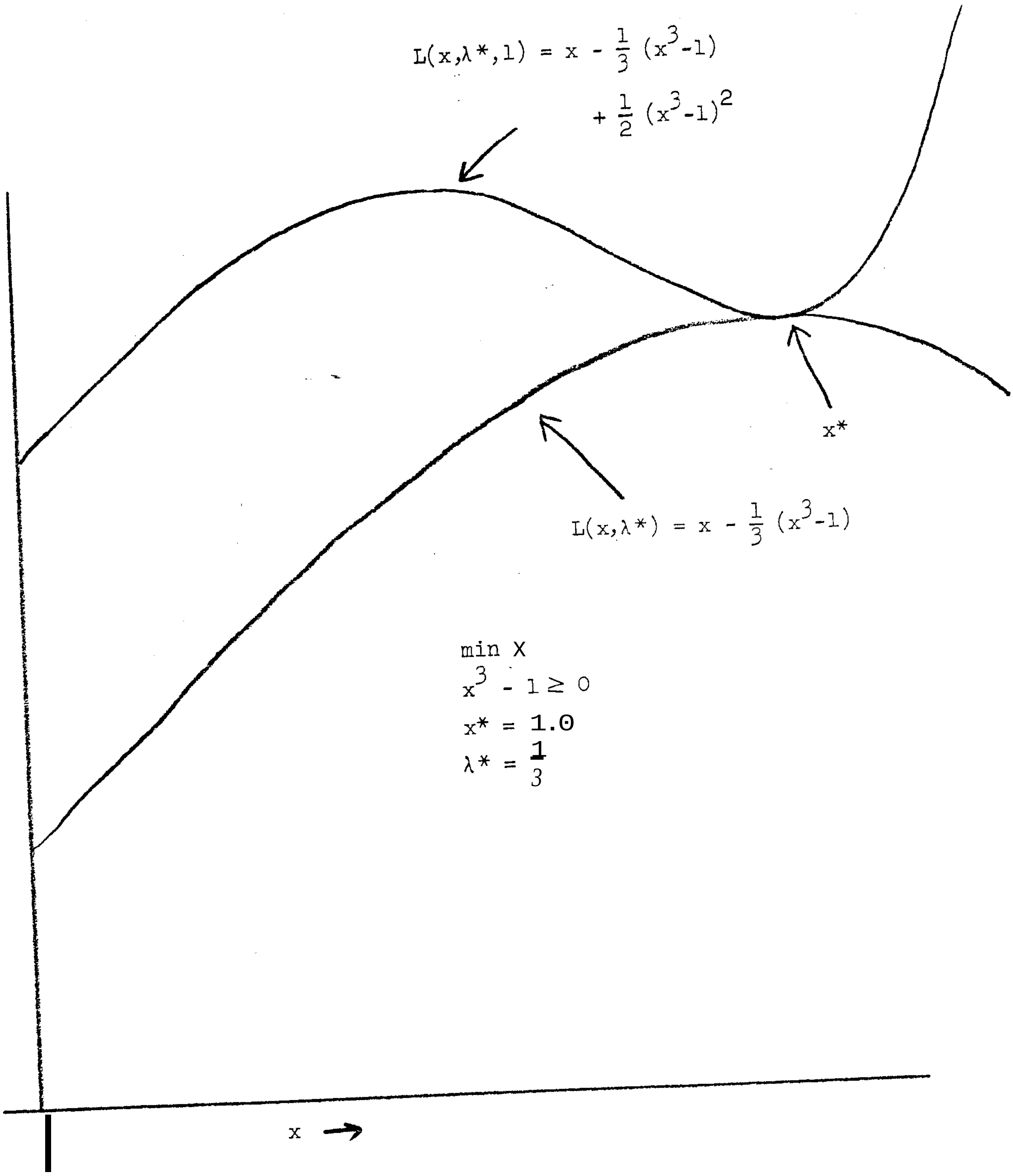

$L(x,\lambda^*,1) = x - \frac{1}{3}(x^3-1)$
$+ \frac{1}{2}(x^3-1)^2$
x^*
$L(x,\lambda^*) = x - \frac{1}{3}(x^3-1)$
min X
$x^3 - 1 \geq 0$
$x^* = 1.0$
$\lambda^* = \frac{1}{3}$
x →

FIGURE 4.1

4.3 Sequential Augmented Lagrangian Methods

~~Hestenes' Method~~

Hestenes (1969) suggested a method for solving P1 involving a sequence of unconstrained subproblems. At each step, the unconstrained minimum of the following function is determined, given a particular choice of the parameter ρ and the vector λ:

$$F - \lambda^T c + \frac{\rho}{2} c^T c. \qquad (4.3.1)$$

The values of λ and ρ are then modified, the unconstrained minimization procedure is repeated for the function (4.3.1) with the new values, and so on, until suitable convergence criteria are satisfied. Clearly (4.3.1) is a special case of the more general augmented Lagrangian function given in Section 4.2, and the procedure suggested by Hestenes can be followed with a penalty term in (4.3.1) of the fcrm $\frac{\rho}{2} c^T Dc$, where D is a positive definite symmetric matrix.

This method will be termed a sequential augmented Lagrangian algorithm; each step cf a sequential algorithm involves finding the unconstrained minimum of an augmented Lagrangian function in which the multiplier vector λ and parameter ρ are kept fixed throughout the minimization, and altered only at the unconstrained minimum. An algcrithmic description of such a method is the following:

```
⟨ Choose an initial vector λ, say $\lambda^{(1)}$, an initial value of ρ,
  say $\rho^{(1)}$, and an initial x, $x^{(0)}$;
     j ← 1; ⟩
 while (the conditions for $x^{(j-1)}$ to be a local minimum of P1
 are not satisfied)
 repeat ⟨
 (1)  Find $x^{(j)}$ [illegible]
```

```
    constrained minimum of $L_A(x,\lambda^{(j)}, \rho^{(j)})$, starting at $x^{(j-1)}$;
    (2) If $||c(x^{(j)})||$ is not sufficiently less than $||c(x^{(j-1)})||$
    determine $\rho^{(j+1)} > \rho^{(j)}$;
    (3) Estimate $\lambda^{(j+1)}$, possibly by updating $\lambda^{(j)}$;
    (4) $j \leftarrow j + 1$;
    〉
```

Powell's Method

Powell (1969) proposed an essentially identical method, but derived it from a different viewpoint, using a quadratic penalty term with a "shift" and weight for each constraint. In Powell's algorithm, the function to be minimized at each stage is given by:

$$F + \frac{1}{2}\sum_{i=1}^{m} \sigma_i(c_i-\theta_i)^2, \tag{4.3.2}$$

where σ_i is the positive weight and θ_i the shift for the ith constraint. The sets of parameters $\{\sigma_i\}$, $\{\theta_i\}$, or both, are altered following each unconstrained minimization.

The function (4.3.2) differs by a constant from the general augmented Lagrangian function $L_A(x,\lambda,\rho)$ given by (4.2.1) in Section 4.2, since (4.3.2) may be written as:

$$F - \sum_{i=1}^{m} c_i\theta_i\sigma_i + \frac{1}{2}\sum_{i=1}^{m} \sigma_i c_i^2 + \frac{1}{2}\sum_{i=1}^{m} \sigma_i\theta_i^2$$

$$= F - \lambda^T c + \frac{\rho}{2} c^T D c + k,$$

where $\lambda_i = \theta_i\sigma_i$, D is a diagonal matrix, diag (d_i), with $d_i = \frac{\sigma_i}{\rho}$, and $k = \frac{1}{2}\theta^T D\theta$. . Hence, Powell's algorithm can be considered as a sequential augmented Lagrangian method.

For simplicity, the remainder of the discussion will deal only with the particular augmented Lagrangian function (4.3.1) proposed by Hestenes, so that the penalty term is identical to the single-parameter quadratic penalty discussed in Chapter 2. However, all results can be applied in a straightforward way to the more general form given initially.

Boundedness of ρ

A feature of the Hestenes and Powell methods is that the parameter representing the penalty for constraint violation -- ρ in (4.3.1), or the values $\{\sigma_i\}$ in (4.3.2) -- need not be increased to infinity to assure convergence of successive unconstrained minima to x^*. Powell showed that each solution $x^*(\lambda,\rho)$ is a local minimum of the constrained problem:

$$\text{minimize } F(x)$$
$$\text{subject to } c_i(x) = c_i(x^*(\lambda,\rho)),\ i=1,2,\ldots,m.$$

Therefore, if the parameters λ and ρ can be chosen so that $c_i(x^*(\lambda,\rho)) = 0$, $i = 1,2,\ldots,m$, the solution of the unconstrained sub-problem will also solve the original problem P1.

The modification of ρ in Step(2)of the sequential augmented algorithm is carried out to assure that the Hessian matrix of $L_A(x,\lambda,\rho)$ is positive definite at the solution, and to enforce an increasing penalty for infeasibility if the multiplier estimates are poor. The hope with an augmented Lagrangian method is that the parameter ρ will eventually (or even initially) exceed the proper threshold, and will not subsequently be altered. If the value of ρ is sufficiently large so that it remains fixed, the explicit dependence of $x^*(\lambda,\rho)$ on ρ may be suppressed, and the solution of a particular unconstrained sub-problem will be written as $x^*(\lambda)$ to emphasize the critical role of the multiplier estimate. Some further aspects of choosing ρ are discussed in Section 4.7.

Restenes/Powell Multiplier Update

The modification of A in Step (3) of the sequential augmented Lagrangian algorithm Is directed toward generatlng an improved estimate of λ^*, since x^* is known to be an unconstrained local ninimum of $L_A(x,\lambda^*,\rho)$ for suitable ρ. The alteration to λ can be carried out in numerous ways, and derived from several perspectives (see Fletcher, 1974).

The update suggested originally by both Hestenes and Powell is:

$$\lambda^{(j+1)} \leftarrow \lambda^{(j)} - \rho^{(j)} c(x^{(j)})$$

where $x^{(j)}$ is the unconstrained minimum of $L_A(x,\lambda(j), \rho^{(j)})$. At $x^{(j)}$, the following holds:

$$\nabla L_A(x^{(j)}, \lambda^{(j)}, \rho^{(j)}) = g - A\lambda^{(j)} + \rho^{(j)} Ac = 0,$$

where all functions are evaluated at $x^{(j)}$, and therefore:

$$g = A(\lambda^{(j)} - \rho^{(j)}c).$$

At x^*, $g = A\lambda^*$, and consequently the updated vector A satisfies at $x^{(j)}$ a relationship of the form that holds at x^*; it is, in addition, a least-squares solution of rain $||g - A\lambda||^2$, where g and A are evaluated at $x^{(j)}$. Powell showed that this update will yield a linear rate of convergence to x^*, with a factor of reduction in error that depends on p.

Other Multiplier Updates

Many other methods for altering the multiplier estimates have been proposed (see Buys, 1972; Fletcher, 1973a), and only one possible form of an update will be given here. The multiplier updates are generally based on the effort to use available information to choose λ to satisfy the conditions that must hold at x^*:

$$g(x^*) = A(x^*)\lambda^* \text{ ; and} \qquad (4.3.3)$$

$$c(x^*) = 0. \qquad (4.3.4)$$

In the following, all vector and matrix functions are evaluated at the current point unless otherwise indicated. If the conctraints are not zero at the current point, the step Ax that must be taken to satisfy (4.3.4) can be estimated by:

$$c(x + \Delta x) = 0$$

$$c(x) + A^{T}\Delta x + O(||\Delta x||^{2}) = 0$$

An approximation $\hat{\Delta}x$ to Ax can be obtained by ignoring all but first-order terms, and specifying:

$${}_{\cdot}{}^{T}\hat{}$$

In order to achieve the relationship (4.3.3), we consider independently the variation in both X and λ, and attempt to characterize Ax and Δλ thst satisfy:

$$g(x + \Delta x) = A(x + Ax)\ (A + \Delta\lambda)$$

$$g + G\Delta x = A\lambda + \sum_{i=1}^{m} \lambda_i G_i \Delta x + A\Delta\lambda + O\ (\max(||\Delta x||^{2},\ ||\Delta\lambda||^{2},$$

If g = AA at the current point, which occurs at the minimum of the augmented function, we obtain an expression relating approximations $\hat{\Delta}x$ and $\hat{\Delta}\lambda$ by ignoring all but first-order terms in Ax and *Ah:*

$$(G - \sum_{i=1}^{m} \lambda_i G_i)\ \hat{\Delta}x = A\ \hat{\Delta}\lambda .$$

If the matrix $G - \sum_{i=1}^{m} \lambda_i G_i$ is nonsingular, this relationship can be written as:

$$\hat{\Delta}x = (G - \sum_{i=1}^{m} \lambda_i G_i)^{-1} A\hat{\Delta}\lambda$$

The condition on $\hat{\Delta}x$ derived earlier is enforced by multiplying both sides by A^{T}, yielding:

$$A^T\hat{\Delta}x = A^T(G - \sum_{i=1}^{m} \lambda_i G_i)^{-1} A\hat{\Delta}\lambda = -c .$$

If $A^T(G - \sum_{i=1}^{m} \lambda_i G_i)^{-1} A$ is nonsingular, the update to the multiplier vector is given by

$$\hat{\Delta}\lambda = - (A^T(G - \sum_{i=1}^{m} \lambda_i G_i)^{-1} A)^{-1} c. \qquad (4.3.5)$$

This correction yields a locally second-order rate of convergence to λ^* if the exact Hessian matrices are available, and the specified assumptions are satisfied. It should be noted that this formulation for the correction $\hat{\Delta}\lambda$ is derived under assumptions much stronger than the sufficient conditions for a minimum given in Section 1.2 (for example, if the matrix $G - \sum_{i=1}^{m} \lambda_i G_i$ is singular at x^* or in a neighborhood of x^*, the given form cannot be used).

Difficulties

The sequential augmented Lagrangian methods can overcome the problem of penalty function methods that the penalty associated with infeasibility must be increased to infinity to assure convergence; however, determination of a suitable value of ρ remains a difficult problem (see Section 4.7).

These methods require repeated unconstrained minimization of a sequence of functions. If the early estimates of λ are inaccurate, or an unfortunate value of ρ is chosen, the sequential-type algorithms may be quite inefficient due to execution of unconstrained minimizations that generate little progress toward the solution.

4.4 Progressive Augmented Lagrangian Methods

An obvious extension of the methods given in Section 4.3 is to update the estimate of the Lagrange multipliers more irequently, rather than keeping λ fixed throughout a complete unconstrained minimization.

Haarhoff and Buys (1970) noted that an accurate minimization of the augmented Lagrangian function witn a ccnstant λ may be unnecessarily lengthy; this idea has since been repeated in varying forms (see Miele et. al., 1971; Tapia, 1975).

A progressive augmented Lagrangian algorithm is defined as a method where an unconstrained minimization technique is applied at each step to minimize an augmented Lagrangian function of the usual form, for given λ and ρ:

$$L_A(x,\lambda,\rho) = F - \lambda^T c + \frac{\rho}{2} c^T c \ .$$

Eowever, the vector λ and the parameter ρ are altered at intervals during the unconstrained minimization, so that the method does not in general attempt to locate the minimum of a fixed function of x from beginning io end. The hope is that this procedure will allow the vector λ to approach λ^* as x approaches x^*.

Such an algorithm is of the following form:

```
    ⟨ Choose an initial $h$, $\lambda^{(1)}$, an initial $\rho$, $\rho^{(1)}$, and an
      initial x, $x^{(1)}$;
      $j \leftarrow 1$; ⟩
      while (the convergence criteria for $x^{(j)}$ to be a local
      minimum of P1 are not satisfied)
      repeat (
(1)   compute a direction of search $p^{(j)}$ and a step $\alpha^{(j)}$, using
      an unconstrained technique to minimize $L_A(x^{(j)},\lambda^{(j)},\rho^{(j)})$;
(2)   $x^{(j+1)} \leftarrow x^{(j)} + \alpha^{(j)} p^{(j)}$;
(3)   compute $\lambda^{(j+1)}$;
(4)   if appropriate, determine $\rho^{(j+1)} > \rho^{(j)}$; otherwise $\rho^{(j+1)} \leftarrow \rho^{(j)}$;
(5)   $j \leftarrow j + 1$;
      ⟩
```

Step (3) need not involve an alteration of λ at every step, so that for some iterations it is possible that $\lambda^{(j+1)} = A(j)$.

The method used in Step (3) to estimate $\lambda^{(j+1)}$ must not require the optimality of $x^{(j)}$ for $L_A(x,\lambda^{(j)},\rho^{(j)})$, and should be selected *so* that $\lambda \rightarrow \lambda^*$ as $x \rightarrow x^*$. A possible choice for the estimate of λ is the vector $\hat{\lambda}$, the least-squares solution of min $||g-A\lambda||2$, for g and A evaluated at the current point; the motivation for this choice is that λ^* satisfies an analogous relationship at x^*. For A of full rank, $\hat{\lambda}$ may be written as $\hat{\lambda} = (A^TA)^{-1}A^Tg$, and will approach λ^* as x approaches x^*. This modification of λ is similar to the original Hestenes/Powell update for sequential algorithms, where the next estimate of λ is an exact least-squares solution. Other estimates of *h* may also be used; for example, a second-order correction corresponding to (4.3.5) may be derived without assuming thst the current point is the unconstrained minimum of the augmented function.

The approach of frequent re-estimation of λ during tha unconstrained minimization procedure is often quite successful in practice (see Chapter 6), and may avoid the inefficiency displayed by the sequential algorithms when the initial estimates of λ are poor. However, a new difficulty is that the criteria for measuring progress toward x^* beccne less apparent, since each unconstrained sub-problem is not solved completely.

An example of this difficulty is the strategy used to choose an appropriate value for p. The sequential algorithms normally increase ρ if there is an insufficient decrease in infeasibility from the unconstrained minimum for a particular *h* to the next minimum. In an algorithm where a new estimate of λ is obtained before the unconstrsined minimum is located for the previous A, it would be unreasonable to expect a decrease in infeasibility at every step. However, the ability to alter ρ must be included in a robust algorithm; hence the decision to increase ρ may involve monitoring the constraint violation over a "reasonable" number of steps, and is not at all straightforward.

4.5 Continuous Augmented Lagrangian Methods

A further possibility for developing a function of which x^* is an unconstrained local minimum is to construct a <u>multiplier</u> <u>function</u> $\lambda(x)$, and create a continuous augmented Lagrangian function which depends only on x and ρ:

$$L_c(x,\rho) = F(x) - \lambda(x)^T c(x) + \frac{\rho}{2} c(x)^T Dc(x),$$

where D is a positive definite symmetric matrix. Such a continuous augmented Lagrangian function was proposed by Fletcher (1970). The dependence of λ on x is the reverse of the analogous relationship in the algorithms described in Sections 4.3 and 4.4: in the sequential methods, esch estimate of x^* depends explicitly on the present estimate of A*; and in the progressive methods, every step in x depends on the current λ, which is fixed during an iteration of the unconstrained sub-problem.

The hope for a continuous augmented Lagrangian method is that x^* will be an unconstrained minimum of $Lc(x,\rho)$ with appropriate choice of p, as for the augmented Lagrangian function given in Section 4.2. Differentlating $L_c(x,\rho)$ with respect to x gives:

$$\nabla L_c(x,\rho) = g - A\lambda - \Lambda c + \rho ADc ,$$

where the <u>i</u>th column of the matrix A is $\frac{\partial c_i}{\partial x}$. At x^*, c=0. In addition, a suitable functional relationship must satisfy $\lim_{x \to x^*} \lambda(x) = \lambda^*$, so that

$g - A\lambda(x^*) = g - A\lambda^* = 0$ because of the sufficient conditions for a minimum at x^*. Therefore, $\nabla L_c(x^*,\rho) = 0$, and x^* is a stationary point of $L_c(x,\rho)$.

The Hessian matrix of $L_c(x,\rho)$ is given by:

$$\nabla^2 L_c(x,\rho) = G - \sum_{i=1}^{m} (A_i G_i + c_i \frac{\partial^2 \lambda_i}{\partial x^2} - \rho v_i G_i) - A\Lambda^T - \Lambda A^T + \rho A D A^T,$$

where $v_i = d_i^T c$ for d_i, the ith column of D. At x^*, $c = 0$ and the Hessian reduces to:

$$\nabla^2 L_c(x^*,\rho) = W - A\Lambda^T - \Lambda A^T + \rho A D A^T.$$

$\nabla^2 L_c(x^*,\rho)$ is positive definite in the null space of $A(x^*)$, since the application of Z^T and Z annihilates the terms involving A and A^T, leaving only $Z^T W Z$, which is positive definite by the sufficient conditions for a minimum at x^*. Any negative curvature of $\nabla^2 L_c$ at x^* must thus occur only along directions in the range of A. By a bounding procedure similar to that given in Section 4.2, it can be shown that there exists a positive value, say $\hat{\rho}$, such that for $\rho > \hat{\rho}$, $\nabla^2 L_c(x^*,\rho)$ is positive definite (see Fletcher, 1970), and x^* is therefore an unconstrained local minimum of the continuous augmented Lagrangian function $L_c(x,\rho)$. This formulation has been called an "exact" penalty function, since a single unconstrained minimization of $L_c(x,\rho)$ will determine x^* for a suitable choice of ρ.

The multiplier function proposed originally by Fletcher is:

$$\lambda(x) = A(x)^+ g(x),$$

where A^+ is the usual pseudo-inverse of A, given by $A^+ = (A^T A)^{-1} A^T$ when A is of full rank (see Peters and Wilkinson, 1970). This definition makes $\lambda(x)$ the least-squares solution of min $||g(x) - A(x)\lambda||^2$ at each point. Other consistent definitions of $\lambda(x)$ are possible (see Fletcher, 1973b).

A notable disadvantage of the continuous approach is that the definition of the Lagrange multipliers in terms of x depends on the gradients of the objective and constraint functions. Second derivatives of the problem functions are consequently required in order to use gradient techniques for unconstrained minimization to minimize $L_C(x,\rho)$. Fletcher (1970) suggests several methods for approximating the derivatives of $\lambda(x)$ without requiring higher derivatives of F and $\{c_i\}$; however, these methods are fairly complicated and have not always been successful in comparison with the methods of Sections 4.3 and 4.4.

A continuous augmented Lagrangian method must also deal with the problem of determining a suitable value of ρ as the calculation proceeds, with the same difficulties described for the methods of Section 4.4 (see Section 4.7 for further discussion).

4.6 Extension to Inequalities

The augmented Lagrangian methods given in Sections 4.3 - 4.5 deal only with equality constraints; we shall now briefly consider their application to the inequality constrained problem:

P2: minimize $F(x)$

subject to $c_i(x) \geq 0$, $i=1,2,\ldots,\ell$.

The sufficient conditions given in Section 1.2 for a minimum of P2 will be assumed to hold at x^*, and $\hat{A}(x^*)$ will be assured to have full rank, where $\hat{A}$ denotes the n by m matrix whose columns are the gradients of the m active constraints at x^*. In particular, these assumptions mean that at x^*, $g = \hat{A}\lambda^*$, $\lambda_i^* \geq 0$. It can be seen immediately that methods based on using properties of the Lsgrangian function at x^* to construct a suitable function of which x^* is an unconstrained minimum must somehow make a determination of the active set of constraints.

Rockafellar (1970, 1973a,b) and Buys (1972) suggested a generalization of the augmented Lagrangian function for the inequality constrained problem, with the following definition:

$$\bar{L}_A(x,\bar{\lambda},\rho) = F + \sum_{i=1}^{\ell} \begin{cases} -\bar{\lambda}_i c_i + \frac{\rho}{2} c_i^2 & \text{if } c_i \le \frac{\bar{\lambda}_i}{\rho} \quad (c_i \text{ is "active"}) \\ -\frac{\bar{\lambda}_i^2}{2\rho} & \text{if } c_i > \frac{\bar{\lambda}_i}{\rho} \quad (c_i \text{ is "inactive"}) \end{cases}$$

As for the augmented function $L_A(x,\lambda,\rho)$ given earlier, this definition can be made more general by including weights for each constraint, but for simplicity, only the definition above will be considered here.

It should be noted that the vector $\bar{\lambda}$ in th[illegible] is an "extended" multiplier vector, since there is a component corresponding to every problem constraint. The general convention used throughout this dissertation (see Section 1.2.3) is that Lagrange multipliers are defined only for the active constraints, and the vector λ^* is always considered to have m components, where there are m active constraints at x^*. However, in this section it may be considered that the "multiplier" corresponding to an inactive constraint is zero at x^*, in order to allow a consistent notation for the augmented Lagrangian methods to be described

The definition of the augmented Lagrangian function $\bar{L}_A$ for inequalities is based on the known non-negativity of the Lagrange multipliers corresponding to constraints active at x^*. At tne unconstrained minimum of $\bar{L}_A(x,\bar{\lambda},\rho)$, $g = \Sigma a_i(\bar{\lambda}_i - \rho c_i)$, where the summation includes only indices for which $c_i \le \frac{\bar{\lambda}_i}{\rho}$. For such an index, the quantity $\bar{\lambda}_i - \rho c_i$ thus serves as an approximation to the Lagrange multi-Flier at x^*, and should be non-negative if c_i is an active constraint. The resulting requirement that $\bar{\lambda}_i - \rho c_i \ge 0$ is precisely the test to determine whether or not the ith constraint is significant in the definition of $\bar{L}_A$. The constant $-\frac{\rho}{2}\bar{\lambda}_i^2$, which is added to $\bar{L}_A$ for a constraint currently considered "inactive", simply makes the function definition continuous at any point where $\bar{\lambda}_i = \rho c_i$.

If c_i is not an active constraint, near x^* it will ultimately make no contribution to the function $\bar{L}_A(x,\bar{\lambda},\rho)$, since c_i will be strictly positive and bounded away from zero, and the corresponding multiplier estimate should approach zero. If c_i is an active constraint, its role in the augmented Lagrangian function is exactly that given for an equality constraint in Section 4.3, with the additional restriction that its multiplier estimate must be non-negative.

An equivalent generalization of the augmented Lagrangian function was proposed by Fletcher (1973a) in the context of Powell's original algorithm, with the function to be minimized given by:

$$F + \frac{1}{2}\sum_{i=1}^{\ell} \sigma_i(\min(c_i-\theta_i,0))^2,$$

where, as before, σ_i is a positive weight and θ_i a shift assigned to the ith constraint. The intention with inequalities is to determine a shift that remains non-zero only for the active constraints; if c_i is active at x^*, the product $\sigma_i\theta_i$ should approach the corresponding optimal multiplier, as in Powell's formulation of the equality case. If $\{\sigma_i\}$, $i=1,2,\ldots,m$, are all equal to ρ, and the relationship $\rho\theta_i = \lambda_i$ holds, the augmented Lagrangian function proposed by Fletcher differs by a constant from the earlier definition of $\bar{L}_A(x,\bar{\lambda},\rho)$.

For either formulation of the augmented Lagrangian function for inequalities, it can be shown (see Buys, 1972) that if $\lambda_i^* > 0$ (all multipliers strictly positive), then there exists $\hat{\rho} > 0$ such that for $\rho > \hat{\rho}$, the solution x^* of P2 is an unconstrained local minimum of $\bar{L}_A(x,\bar{\lambda}^*,\rho)$, where the components of the extended vector $\bar{\lambda}^*$ are defined in terms of whether the corresponding constraint is active at x^*:

$\bar{\lambda}_i^* = \lambda_{\hat{i}}^*$ if the ith constraint is active, corresponding to index $\hat{i}$ in the set I of active constraints;

$\bar{\lambda}_i^* = 0$ if the ith constraint is inactive at x^*.

The augmented function $\bar{L}_A(x,\bar{\lambda},\rho)$ contains discontinuities in the second derivative at any point where $\lambda_i = \rho c_i$, but the hope is that such points will occur far from x^* and hence have little effect on the success of whatever unconstrained algorithm is used to minimize L_A.

A sequential-type algorithm -- in which successive unconstrained minima of $\bar{L}A(x,\bar{\lambda},\rho)$ are found for fixed $\bar{\lambda}$ -- can be applied to an inequality constrained problem, and the test as to which constraints are included in the summation at any point is an implicit decision about the active set. However, the problem of updating $\bar{\lambda}$ at the end of each unconstrained minimization is more complicated than for the equality problem.

The analogue of the originel Hestenes/Powell correction is:

$$\Delta\bar{\lambda}_i = - \min\,(\rho c_i,\bar{\lambda}_i) \ .$$

If the original $\{\bar{\lambda}_i\}$ are all non-negative, the correction for any "active" constraint (for which $\bar{\lambda}_i \geq \rho c_i$) will be $-\rho c_i$, and the updated $\bar{\lambda}_i$ will remain non-negative. The correction for any "inactive" constraint (for which $\bar{\lambda}_i < \rho c_i$) will be $-\bar{\lambda}_i$, so that the updated multiplier estimate will be zero. Such an "inactive" constraint will be considered "active" at a subsequent point only if c_i has been violated, to satisfy $\rho c_i < 0$; if c_i

remains violated at the next minimum of $\bar{L}_A(x,\bar{\lambda},\rho)$, the updated $\bar{\lambda}_i$ will again be positive. Hence, the Hestenes/Powell-tvpe updating procedure assures non-negative multiplier estinatcs if the original estimates are nor,-negative.

The guarantee of non-negativity for multiplier estimates corresponding to active constraints does not necessarily apply to other updates of $\bar{\lambda}$, and may need to be imposed on the updating method. Fletcher (1973a) suggests some possible approaches to obtaining higher-order estimetes of $\bar{\lambda}$ which assure that all estimates are non-negative.

The progressive methods described in Section 4.4 may also be extendea to the inequality case (see Buys, 1972), but the two related problems of determining the current "active set" and estimating the multipliers at each iteration can become quite complicated. For example, the estimate of λ analogous to the least-squares solution described in Section 4.4 is entirely dependent on the chosen specification of the active constraints, and the multipliers are not guaranteed to be positive. A set of non-negative multipliers can be found by solving:

$$\min \ ||g-\tilde{A}\lambda||^2$$

$$\text{subject to } \lambda \geq 0,$$

where the matrix $\tilde{A}$ might include only the conjectured "active set" or possibly all the problem constraints; but this calculation may be length;., and the validity of the resulting multipliers may be questionable. If non-negativity is not imposed on the estimates of A, a suitable strategy must be devised to deal with any constraint with a negative multiplier *estimate.*

The continuous augmented Lagrangian methods can be applied to the inequality problem by using an "active set'' strategy for every evaluation of the function $L_c(x,\rho)$ (see Lill, 1972); however, this approach contains the undesirable feature that any change in the active set introduces discontinuities in the function being minimized. To overcome this disadvantage, Fletcher (1973a) has defined the multiplier function for the inequality problem as the solution of an inequality constrained quadratic programming problem. Eowever, the complications mentioned earlier for the continuous methods apply in either case.

4.7 Discussion of Unconstrained Lagrangian Methods

The methods described in Sections 4.3 - 4.6 convert an original constrained minimization problem into a finite sequence (possibly of length one) of unconstrained sub-problems. These methods can in theory overcome the ill-conditioning associated with penalty functions by generating a progressively more accurate estimate of the Lagrange multipliers at x^*, so that **it** is possible *to* avoid the necessity of increasiog to infinity the penalty for infeasibility. For many problems, the unconstrained augmented Lagrangian algorithms have been highly successful, particularly the sequential and progressive methods (see Chapter 6).

Unboundedness

The danger that the unconstrained function to be minimized is unbounded below, which can occur with a penalty function formulation, clearly exists for these methods as well, and any unconstrained algorithm to he applied to an augmented Lagrangian function should be able to recover from this'situation. Since the penalty parameter is not necessarily increased after every unconstrained minimization, the risk of unboundedness may be greater for the Lagrangian-type methods.

Choice of Parameter ρ

The necessity to choose a parameter ρ occurs in all the Lagrangian algorithms discussed thus far. Part of the motivation for development of these algorithms was to allow convergence to x^* for a finite value of ρ, in contrast to penalty function methods. However, in general ρ must still be bounded below, since x^* will be an unconstrained local minimum of $L_A(x,\lambda^*,\rho)$ only if ρ is chosen to exceed a threshold, say $\hat{\rho}$, such that the Hessian matrix of $L_A(x^*,\lambda^*,\hat{\rho})$ is positive definite.

The n eigenvalues of the matrix $\nabla^2 L_A(x^*,\lambda^*,\rho) = W + \rho AA^T$, where W is the Hessian of the Lagrangian function, can be considered as a function of ρ, since W and A remain fixed; let $\mu_1(\rho) \leq \mu_2(\rho) \leq \ldots \leq \mu_n(\rho)$ denote the ordered set of eigenvalues of $W + \rho AA^T$. For $\nabla^2 L_A(x^*,\lambda^*,\rho)$ to be positive definite, $\mu_1(\rho)$ must be strictly positive. Let $\eta(\rho)$ be the spectral condition number of $\nabla^2 L(x^*,\lambda^*,\rho)$, defined as:

$$\eta(\rho) = \frac{\max_i |\mu_i(\rho)|}{\min_i |\mu_i(\rho)|} .$$

Figure 4.2 displays a typical plot of the variation in $\eta(\rho)$. The number of poles in $\eta(\rho)$ is equal to the number of negative eigenvalues of

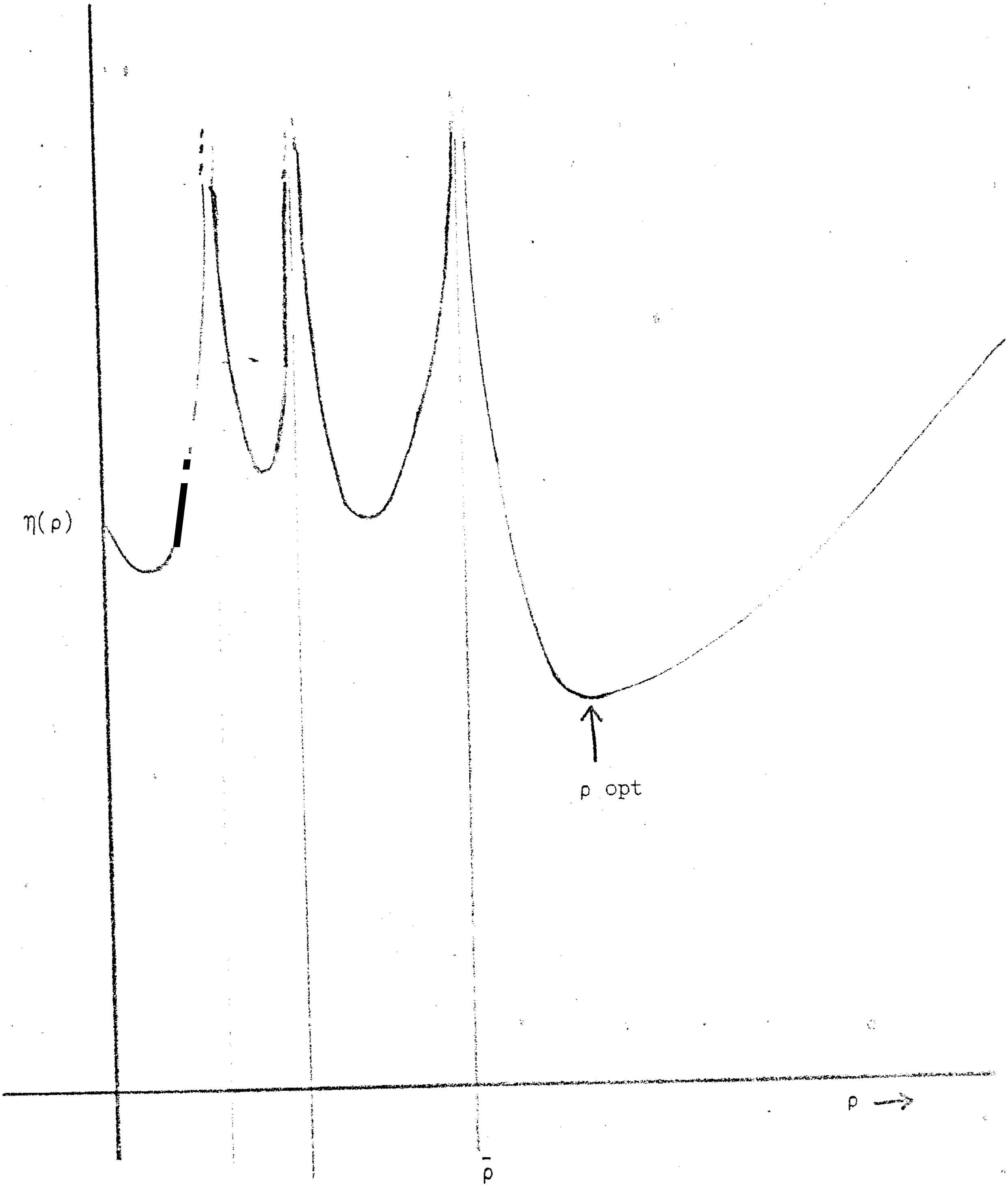


FIGURE 4.2

$W(x^*)$, and each pole corresponds to a value of p that shifts an original negative eigenvalue of W to zero.

Asymptotically, the behavior of $\eta(\rho)$ for large ρ has keen studied in detail by Murray (1971), and his analysis reveals that

$\lim\limits_{\rho \to \infty} \frac{\quad}{P} = 1$, so that the curve in Figure 4.2 approaches a constant 45° as $\rho \to \infty$. **It** is clear that a "large" value of ρ will cause ill-conditioning of the matrix $W + \rho AA^T$; hcwever, the definition of "large" may vary dramatically from problem to problem because a "large" value of ρ may, in some cases, be required to make $W + \rho AA^T$ positive definite. A further, perhaps more interesting, feature of Figure 4.2 is thot there is danger of singularity or ill-conditioning of the matrix $W + \rho AA^T$ when the value of ρ is "too small", since there may be little variation between a value of ρ for which $W + \rho AA^T$ is singular or close to singular, and tlie value ρ_{opt} for which $W + \rho AA^T$ is positive definite and the spectral condition number is smallest.

The method used to determine a suitable p for an augmented Lagrangian algorithm may, therefore, have difficulty if an effort is made to choose the smallest possible value of p. Since the quantities required to specify a lower bound for p are based on functions evaluated at (x^*,λ^*), the estimate of the bound will be based on similar quantities obtained at some other point (x,λ), and may be quite inaccurate. For algorithms where ρ is kept fixed for several iterations, or for an entire unconstrained minimization, an unfortunate choice of p may lead to considerable inefficiency.

Rate of Convergence

The rate at which the sequence of iterates generated by an augmented Lagrangian method converges to x^* is critically dependent on the accuracy of the current multiplier estimate. Even if a second-order method is used to obtain the step at each iteration, convergence to x^* will cnly be linear if the convergence of the multipliers to λ^* is linear.

This result will be demonstrated for the case when exact second derivatives of F and $\{c_i\}$ are available. The point $\bar{x}$ is the current estimate of x^*, with a similar meaning for $\bar{\lambda}$ The step $\bar{p}$ to be computed is the Newton step to the minimum of the augmented function $L_A(x,\bar{\lambda},\rho) = F - \bar{\lambda}^T c + \frac{\rho}{2} c^T c$ (p is assumed to be fixed and bounded), and is defined by:

$$\nabla^2 L_A(x,\bar{\lambda},\rho)| \quad \bar{p} = - \nabla L_A(x,\bar{\lambda},\rho)| \qquad (4.7.1)$$

The next estimate of the solution is $\bar{\bar{x}} = \bar{x} + \bar{p}$. Let $\bar{e} = \bar{x} - x^*$, $\bar{\bar{e}} = \bar{\bar{x}} - x^*$, and $\bar{\Delta} = \bar{\lambda} - \lambda^*$, so that $\bar{p} = \bar{\bar{e}} - \bar{e}$. Expanding about $\bar{x}$, we obtain the following estimates:

$$\text{(a)} \quad \bar{g} - \bar{G}\bar{e} = g(x^*) + O(||e||^2);$$

$$\text{(b)} \quad - \bar{A}^T\bar{e} = c(x^*) + O(||\bar{e}||^2) = O(||\bar{e}||^2);$$

$$\text{(c)} \quad \bar{a}_i - \bar{G}_i\bar{e} = a_i(x^*) + O(||\bar{e}||^2),$$

where the notation "$\bar{\alpha}$" means the function "α" evaluated at $\bar{x}$.

In (4.7.1), substituting the expression $\bar{p} = \bar{\bar{e}} - \bar{e}$, writing out the elements of $\nabla^2 L_A$ multiplying $\bar{e}$, and re-grouping, we obtain:

$$\nabla^2 L_A \bar{\bar{e}} = (-\bar{g}+\bar{G}\bar{e}) - \rho\bar{A}(\bar{c}-\bar{A}^T\bar{e}) + (\bar{A}\bar{\lambda} - \sum^m \bar{\lambda}_i \bar{G}_i \bar{e}) + \sum^m \rho\bar{c}_i \bar{G}_i \bar{e} \ .$$

Substituting from (a), (b), end (c) in the first, second, and third expressions in parentheses, respectively, gives:

$$\nabla^2 L_A \bar{\bar{e}} = -g(x^*)+A(x^*)\bar{\lambda} + \sum^{m} \rho \bar{c}_i \bar{G}_i \bar{e} + O(||\bar{e}||^2),$$

where the boundedness of ρ is required in order to group some of the quantities in the $O(||\bar{e}||^2)$ term.

For sufficiently small $||\bar{e}||$, $||\bar{c}||$ will be of order $||\bar{e}||$, so that the term involving $\bar{c}_i$ and $\bar{e}$ can also be included as $O(||\bar{e}||^2)$, again because ρ is bounded.

The resulting relationship is:

$$\nabla^2 L_A \bar{\bar{e}} = -g(x^*)+A(x^*)\bar{\lambda} + O(||\bar{e}||^2).$$

Substituting $\bar{\lambda} = \lambda^* + \bar{\Delta}$ gives:

$$\nabla^2 L_A \bar{\bar{e}} = -g(x^*)+A(x^*)\lambda^* + A(x^*)\bar{\Delta} + O(||\bar{e}||^2), \text{ or}$$

$$\nabla^2 L_A \bar{\bar{e}} = A(x^*)\bar{\Delta} + O(||\bar{e}||^2), \qquad (4.7.2)$$

since $g(x^*) - A(x^*)\lambda^* = 0$. Because $\nabla^2 L_A$ is assumed to be nonsingular, the relationship (4.7.2) means t $|\bar{\Delta}||$, $||\bar{e}||^2)$ and is therefore limite the error in $\bar{\lambda}$. This result will be illustrated by numerical examples in Chapter 6.

4.8 Linearly Constrained Lagrangian Methods

4.8.1 Introduction

The problem to be considered first is the equality constrained problem, P1. The augmented Lagrangian algorithms discussed in Sections 4.3 - 4.6 were based on constructing a function of which the constrained solution x* is an unconstrained minimum. Because the Hessian of the Lagrangian function may be indefinite in the subspace spanned by the columns of A(x*), a penaity term must be included in the augmented Lagrangian function in order to add sufficient positive curvature in this subspace to make the Hessian of the augmented function positive definite at x*. The resulting necessity to choose and adjust a penalty parameter p creates difficult problems for the augmented methods.

An alternative possibility for the sub-problem satisfied by x^* is a linearly constrained minimization problem based on the Lagrangian function. With a suitable choice of linear constraints, it may be possible to restrict the subspace in which the minimization occurs to that in which the Hessian of the Lagrangian function is positive definite at x^*. Such a formulation would eliminate the critical role of the penalty term in assuring that x^* is a solution of the sub-problem.

The sufficient conditions given in Section 1.2 for x^* to be a constrained minimum of P1 include:

(1) $c(x^*) = 0$;

(2) $g(x^*) = A(x^*)\lambda^*$, or, equivalently, $Z(x^*)^T g(x^*) = 0$;

(3) $p^T W(x^*)\, p > 0$ for any $\mathbf{p}$ such that $A(x^*)^T p = 0$,

or, equivalently, $Z(x^*)^T W(x^*) Z(x^*)$ is positive definite, where

$$W(x^*) \equiv G(x^*) - \sum^{m} \lambda_i^* G_i(x^*$$

The linearly constrained sub-problem to be considered is of the form:

LCP: minimize $fL(x)$

subject to $A_L^T x = d_L$.

The sufficient conditions that x^* be a minimum of the linearly constrained prcblem LCP are:

(1') . $A_L^T x^* = d_L$;

(2') $\nabla f_L(x^*) = A_L \mu$ for some vector μ, or, equivalently,

$Z_L^T \nabla f_L(x^*) = 0$, where the columns of Z_L form an orthogonal basis for the null. space of the columns of A_L;

(3') $Z_L^T \nabla^2 f_L(x^*) Z_L$ is positive definite.

A beneficial feature of the formulation LCP might be that the multipliers μ of the linearly constrained problem (satisfying condition (2') at the solution) are related to the Lagrange multipliers λ^* of the original problem P1. However, this property is not required for x^* to solve the sub-problem.

4.8.2 Design of Linearly Constrained Sub-Problem

Clearly a sub-problem LCP solved by x^* could be formulated if values at x^*, such as λ^* and $A(x^*)$, were used to construct the function and set of linear constraints (in exactly the same way as an augmented Lagrangian function cf which x^* is an unconstrained minimum can be defined if λ^* and a value of ρ, depending on x^*, are known a priori). However, because x^* is unknown, the definition of the sub-problem LCP will depend on the current estimate of the solution; the function and constraints of LCP should be specified so that x^* will satisfy conditions (1'), (2'), and (3') if the sub-problem happens to be defined at x^*.

There are many possible specifications of the sub-problem LCP, and only one will be given here.

Let $\bar{x}$ be the current estimate of x^*, with $\bar{\lambda}$ the current multiplier estimate; $\bar{A}$ and $\bar{c}$ denote $A(\bar{x})$ and $c(\bar{x})$, respectively. The particular formulation of LCP to be considered depends on both $\bar{x}$ and $\bar{\lambda}$, and will be written as LCP $(\bar{x},\bar{\lambda})$.

LCP($\bar x,\bar\lambda$): minimize $F(x) - \bar\lambda^T c(x) + \bar\lambda^T \bar A^T x$

subject to $\bar A^T x = -\bar c + \bar A^T \bar x$.

In terms of the earlier definition of LCP, $A_L \equiv A$, d_L - $-\bar c + \bar A^T \bar x$, and $f_L(x) \equiv F(x) - \bar\lambda^T c(x) + \bar\lambda^T \bar A^T x$; note that $\nabla f_L(x) = g(x) - A(x)\bar\lambda + \bar A\bar\lambda$, and $\nabla^2 f_L(x) = G(x) - \sum_{i=1}^{m} \bar\lambda_i G_i(x)$.

The constrained solution x^* satisfies the conditions for a minima of LCP(x^*,λ^*):

(1) $A_L^T x^* = A(x^*)^T x^* = d_L = -c(x^*) + A(x^*)^T x^*$, since $c(x^*) = 0$;

(2) $\nabla f_L(x^*) = g(x^*) - A(x^*)\lambda^* + A(x^*)\lambda^* = g(x^*) = A(x^*)\lambda^*$;

(3) $Z_L^T \nabla^2 f_L(x^*) Z_L$ is positive definite, since

$Z_L = Z(x^*)$ and $\nabla^2 f_L(x^*) = W(x^*)$.

Furthermore, the Lagrange multiplier vector Gf LCP(x^*,λ^*) is λ^*, the multiplier vector of the original problem.

The linear constraints of LCP($\bar x,\bar\lambda$) specify a step, p, to be taken from $\bar x$ that satisfies $\bar A^T p = -\bar c$. These constraints are a linearization of the constraint functions about $\bar x$, and give a first-order prediction, exact for linear constraints, that the constraint: will be zero at the next point. Even for nonlinear constraints, p is a descent direction for the function $c^T c$, since $p^T \nabla (c^T c)|_{\bar x} = 2p^T \bar A \bar c = -2\bar c^T \bar c$ C 0. However, the full step p will not necessarily reduce infeasibility with respect to the original nonlinear constraints (see Section 4.8.7).

4.8.3 Sequential Linearly Constrained Lagrangian Methods

Robinson (1972) and Rosen and Kreuser (1972) have proposed algorithms based on solution of a sequence of linearly constrained sub-problems closely related to the given formulation of LCP($\bar{x},\bar{\lambda}$). A sequential algorithm based on formulation is:

```
(Given an initial x, say $x^{(0)}$, and an initial $\lambda$, say $\lambda^{(0)}$;
 j ← 1;⟩
 while (the conditions for $x^{(j-1)}$ to be a solution of P1
 are not satisfied)
 repeat (
     (1)  starting at $x^{(j-1)}$, find the solution $x^{(j)}$ and
          the multiplier vector $\lambda^{(j)}$ of the linearly ccn-
          strained problem LCP($x^{(j-1)}, \lambda^{(j-1)}$);
     (2)  j ← j + 1;
     ⟩
```

Robinson (1972) presented a proof that under certain conditions on x*, if ($x^{(0)},\lambda^{(0)}$) is sufficiently close to (x^*,λ^*), the sequence $\{x^{(j)},\lambda^{(j)}, j=1,2,\ldots\}$ converges quadratically to (x^*,λ^*). Some aspects of the convergence of this algorithm will be discussed in Section 4.8.5. It should be noted that the step from ($x^{(j-1)},\lambda^{(j-1)}$) to ($x^{(j)},\lambda^{(j)}$) involves the solution of a general linearly constrained sub-problem, and tne quadratic rate of convergence, therefore, applies to a rather broad definition cf "step".

A sequential linearly constrained method suffers from the same possible defect mentioned for sequential augmented Lagrangian methods, that the algorithm may be inefficient because of the effort required to solve completely a sub-problem which has relevance tc the originel problem only close to x*. Other difficulties will be considered in Section 4.8.7.

4.8.4 Progressive Linearly Constrained Lagrangian Methods

A progressive approach, similar to that described in Section 4.4 for the augmented methods, may be taken for the linearly constrained Lagrangian methods, in order to overcome possible inefficiency of the sequential algorithms. In a progressive method, the sub-problem to be solved (approximately) at the next iteration is modified, based on information acquired during steps toward the solution of the previous sub-problem. Only one progressive algorithm will be discussed here, to highlight the primary strategy involved.

If the function f_L in the linearly constrained sub-problem $LCP(\bar{x},\bar{\lambda})$ were a quadratic with a known Hessian, the sub-problem would reduce to a quadratic programming problem with equality constraints, whose sclution can under certain conditions be explicitly written down (see Gill and Murray, 1974b). A progressive method could thus be based on making a quadratic approximation to the function f_L at every point, and solving the resulting quadratic programming problem. The quadratic programming problem associated with $LCP(\bar{x},\bar{\lambda})$, can be conveniently written in terms of the step p from $\bar{x}$ to the next point, $\bar{x}$, as follows:

$$QP(\bar{x},\bar{\lambda}): \qquad \underset{p}{\text{minimize}}\ \tfrac{1}{2}p^T\bar{S}p + p^T\bar{g}$$
$$\text{subject to } \bar{A}^Tp = -\bar{c}\ ,$$

where $\bar{S}$ is a local approximation to the Hessian of the Lagrangian function, based on $\bar{x}$ and $\bar{\lambda}$, and where $\bar{g}$, $\bar{A}$, and $\bar{c}$ denote $g(\bar{x})$, $A(\bar{x})$ and $c(\bar{x})$, respectively.

This type of progressive algorithm was suggested by Wilson (1963), and may be described 3s follows:

```
(Given an initial x, say $x^{(1)}$, and an initial $\lambda$, say
$\lambda^{(1)}$;
    $j \leftarrow 1$;⟩
while (the conditions for $x^{(j-1)}$ to be a minimum of P1
        are not satisfied)
repeat ⟨
    (1) Compute the solution $p^{(j)}$ of $QP(x^{(j)},\lambda^{(j)})$;
    (2) $x^{(j+1)} \leftarrow x^{(j)} + p^{(j)}$;
        [illegible]
        which satisfy:
            $A^{(j)}\lambda^{(j+1)} = g^{(j)} + S^{(j)}p^{(j)}$,
        where "$\alpha^{(j)}$" denotes "$\alpha$" evaluated at $x^{(j)}$;
    (4) $j \leftarrow j + 1$;
⟩
```

It can be shown (see Robinson, 1973) that if tne exact Hessian matrices of F and $\{c_i\}$ are used to calculate $S^{(j)}$, so that $S^{(j)} = G^{(j)} - \sum_{i=1}^{m} \lambda_i^{(j)} G_i^{(j)}$, and if $(x^{(1)},\lambda^{(1)})$ is sufficiently close to (x^*,λ^*), then the progressive algorithm will display quadratic convergence to x^*. Additional comments on the rate of convergence are given in Section 4.8.5.

4.a.5 Convergence of Linearly Constrained Lagrangian Alg or ithms

For an initial $(\bar{x},\bar{\lambda})$ sufficiently close to (x^*,λ^*), a quadratic rate of convergence can be proved for either the sequential methods, or progressive methods where exact second derivatives are available (see Robinson, 1973). These cases have essentially identical prnofs; thus, the arialysis will be given for a progressive method based on the quadratic programming sub-problem $QP(\bar{x},\bar{\lambda})$, where the exact second derivatives are available.

The proof is of interest for the following reason. With an augmented Lagrangian algorithm, even one which uses a second-order method to solve the unconstrained sub-problem, the rate of convergence to x^* is defined by the rate of convergence of the multiplier estimates (see Section 4.7). However, this limitation does not apply to the given formulation of linearly constrained methods; hence, a first-order estimate of the multipliers need not prevent second-order convergence to x^*.

Let $\bar{x}$ and $\bar{\lambda}$ be the current estimates of x^* and λ^*. The step $\bar{p}$ to the next estimate, $\bar{\bar{x}}$, is the solution of the quadratic programming problem:

$$\text{minimize } \frac{1}{2} p^T(\bar{G} - \sum_{i=1}^{m} \bar{\lambda}_i \bar{G}_i)p + p^T\bar{g}$$

$$\text{subject to } \bar{A}^T p = -\bar{c},$$

where "$\bar{\alpha}$" denotes "α" evaluated at $\bar{x}$. The new multiplier estimate $\bar{\bar{\lambda}}$ will be given by the multipliers of the quadratic programming problem.

Define $\bar{e} = \bar{x}-x^*$, $\bar{\bar{e}} = \bar{\bar{x}}-x^*$, $\bar{\Delta} = \bar{\lambda}-\lambda^*$, $\bar{\bar{\Delta}} = \bar{\bar{\lambda}}-\lambda^*$.

We then make the following estimates by expanding about $\bar{x}$:

(a) $\bar{c}-\bar{A}^T\bar{e} = c(x^*) + O(||\bar{e}||^2) = O(||\bar{e}||^2)$;

(b) $\bar{g}-\bar{G}\bar{e} = g(x^*) + O(||\bar{e}||^2)$;

(c) $\bar{a}_i-\bar{G}_i\bar{e} = a_i(x^*) + O(||\bar{e}||^2)$.

Since $\bar{p}$ satisfies $\bar{A}^T\bar{p} = -\bar{c}$, by writing $\bar{p}$ as $\bar{\bar{e}}-\bar{e}$, we obtain:

$$\bar{A}^T\bar{\bar{e}} = -\bar{c} + \bar{A}^T\bar{e} .$$

Substituting from (a), this relationship becomes:

$$\bar{A}^T\bar{\bar{e}} = O(||\bar{e}||^2). \qquad (4.81)$$

The estimate $\bar{\bar{\lambda}}$ is defined by:

Again, substituting $\bar{\bar{e}}-\bar{e}$ for $\bar{p}$, the result is:

$$\bar{A}\bar{\bar{\lambda}} = \bar{g} + (\bar{G} - \sum_{i=1} \bar{\lambda}_i\bar{G}_i)\bar{\bar{e}} - \bar{G}\bar{e} + \sum_{i=1} \bar{\lambda}_i\bar{G}_i\bar{e} \; .$$

Defining $\bar{W}$ to be $(\bar{G} - \sum_{i=1}^{m} \bar{\lambda}_i\bar{G}_i)$, and substituting from (b) gives:

$$\bar{A}\bar{\bar{\lambda}} = \bar{W}\bar{\bar{e}} + g(x^*) + \sum_{i=1}^{m} \bar{\lambda}_i\bar{G}_i\bar{e} + O(||\bar{e}||^2).$$

Writing $\bar{\bar{\lambda}}$ as $\lambda^* + \bar{\bar{\Delta}}$, and $\bar{\lambda}$ as $\lambda^* + \bar{\Delta}$, gives:

$$\bar{A}\bar{\bar{\Delta}} = \bar{W}\bar{\bar{e}} + g(x^*) - \bar{A}\lambda^* + \sum_{i=1}^{m} \lambda^*_i\bar{G}_i\bar{e} + \sum_{i=1}^{m} \bar{\Delta}_i\bar{G}_i\bar{e} + O(||\bar{e}||^2).$$

Substituting from (c), we obtain:

$$\bar{A}\bar{\bar{\Delta}} = \bar{W}\bar{\bar{e}} + g(x^*) - A(x^*)\lambda^* + \sum_{i=1}^{m} \bar{\Delta}_i\bar{G}_i\bar{e} + O(||\bar{e}||^2),$$

and, since $g(x^*) = A(x^*)\lambda^*$, the result is:

[illegible] (4.8.2)

The derived relationships (4.8.1) and (4.8.2) for $\bar{\bar{e}}$ and $\bar{\bar{\Delta}}$ may then be written:

$$\begin{bmatrix} \bar{W} & -\bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} \bar{\bar{e}} \\ \bar{\bar{\Delta}} \end{bmatrix} = \begin{bmatrix} O(||\bar{\Delta}|| \; ||\bar{e}||) + O(||\bar{e}||^2) \\ O(||\bar{e}||^2) \end{bmatrix} . \qquad (4.8.3)$$

If $A(x^*)$ has full rank, then for $\bar{x}$ sufficiently close to x^*, the matrix on the left-hand side of (4.8.3) is non-singular. The form of the right-hand side shows the second-order convergence, since if $O(||\bar{e}||) = O(||\bar{\Delta}||)$, both x and λ converge quadratically to (x^*,λ^*). This property indicates that close to x^* the convergence of the linearly constrained methods is not limited by first-order. errors in the Lagrange multiplier estimates.

4.8.6 Inequality Constraints

The formulation of a linearly constrained sub-problem for the inequality constrained problem, P2, is analogous to the equality .case, except that the linear constraints of the corresponding sub-problem are inequaiities. Consequently, the approaches discussed previously for equalities may be applied to the inequality problem by changing the constraints of LCP to the form:

$$A_L^T x \geq d_L \quad .$$

An inactive constraint of the linearly constrained sub-problem should ideally correspond to an inactive constraint of the nonlinear problem, so that properties of the solution of LCP or QP can be used to provide information about the solution of P2. If a linear constraint of the sub-problem is active, its multiplier will of necessity be non-negative, and is associated with a non-negative estimate for a multiplier of the original problem; similarly, when a linear constraint is inactive, the predicted multiplier for the corresponding nonlinear constraint may be taken as zero. Hence, the relationship of the linearized constraints to those of the original problem becomes critical.

4.8.7 Discussion of Linearly Constrained Lagrangian Methods

The methods based on formulating a linearly constrained sub-problem involving the Lagrangian function are highly successful close to the solution, and avoid the necessity of dealing with a penalty term whose only purpose is to shift negative eigenvalues of $W(x^*)$. However, the sub-problem to be solved has become more complicated, particularly in the case of inequality Constraints.

A more fundamental difficulty is that the relationship between the solution of the linearly constrained sub-problem and that of the original problem is questionable when the initial estimates of x^* and λ^* are far from optimal. For example, at any stage the sub-problem may be unbounded below. Even if the numerical technique used to solve the sub-problem can detect this situation, the subsequent definition of the algorithm is unclear; with an augmented algorithm, the ability to increase the penalty parameter could overcome this problem. Furthermore, there is no guarantee of an ultimate decrease in infeasibility with the linearly constrained methods as presented here; nor is the meaning obvious of an approximation to the Lagrangian function at a point where $||c||$ is large, or $\bar{\lambda}$ differs widely from λ^*.

Safeguards must be included in linearly constrained Lagrangian algorithms in order to assure that some measure of progress toward x^* is satisfied. These methods thus require further study and refinement to make them robust and of general usefulness.

CHAPTER 5. Methods Based on the Trajectory of Penalty arid Barrier Functions

CHAPTER 5

METHODS BASED ON THE TRAJECTORY OF PENALTY AND BARRIER FUNCTIONS

5.1 Introduction

.The Lagrangian methods described in Chapter 4 overcome only some of the deficiencies ass'ociated with penalty and barrier function methods. Moreover, their design includes additional levels of complication and uncertainty. Their theoretical superiority is demonstrated only close to the solution, and critically depends on a suitable choice of various parameters; away from the solution, where the local conditions for a minimum are not satisfied, such methods can fail to make any progress toward the solution without reverting to use of a penalty-type method. For a problem where strict feasibility must be maintained throughout, the Lagrangian-type methods cannot be applied directly, since they do not theoretically restrict the sequence of estimates of the solution to be feasible.

The algorithms to be discussed in this chapter are based on an effort to overcome the difficulties associated with penalty and barrier functions without incurring additional problems . This approach was first developed by Murray (1969a,b), who presented a detailed algorithm for the penalty function case. These "trajectory" algorithms are designed to exploit the following properties of penalty and 'carrier function methods: (i)the characteristic non-tangential approach to the solution that justifies the use of linear approximations to the constraint functions even close to the solution; (2) a straightforward set of criteria to measure progress at each iteration; (3) variation of' a single parametnr to control convergence; (4) a special relationship between the values of the constraint functions and the estimates of the Lagrange multipliers; (5) in the barrier function case, maintenance of feasibility in the seqaence of estimates of the solution.

5.2 Further Properties of the Quadratic Penalty Function

Equality Constraints

The initial discussion in this section will be concerned with the solution of the equality constrained problem, P1, by use of the quadratic penalty function first described in Chapter 2:

$$P(x,\rho) = F(x) + \frac{\rho}{2} c(x)^T c(x).$$

We shall assume the following: the sufficiency conditions giver? in Section 1.2 for a minimum of P1 hold at x*, A(x*) is of full rank, $\lambda_i^* \neq 0$ for any i, $||G(x^*)||$ and $||G_i(x^*)||$, i=1,2,...,m, are bounded, and $\lim_{\rho \to \infty} x^*(\rho) = x^*$.

At the minimum of P(x,ρ):

$$\nabla P(x^*(\rho),\rho) = g + \rho Ac = 0. \qquad (5.2.1)$$

Consider the perturbation $\Delta x = x^*(\bar\rho) - x^*(\rho)$, where $\bar\rho > \rho$ and $\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)$ is

$$g(x^*(\bar\rho)) = - \bar\rho A(x^*(\bar\rho))\, c(x^*(\bar\rho)), \text{ or}$$

$$g(x^*(\rho) + \Delta x) = -\bar\rho A(x^*(\rho) + \Delta x)\, c(x^*(\rho) + \Delta x).$$

Expanding g, A, and c in their Taylor series about x*(ρ) yields:

$$g + G\Delta x = - \bar\rho Ac - \bar\rho AA^T\Delta x - \sum^{m} \bar\rho c_i(x^*(\bar\rho))G_i\Delta x + O(||\Delta x||^2)$$

We now estimate the size of each term. Because of the properties of the trajectory defined by minima of P(x,ρ) (see Section 2.2.3), $||\Delta x|| = O\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)$. $||G||$ and $||G_i||$ are Sounded at x*, so that $||G(x^*(\rho))||$ and $||G_i(x^*(\rho))||$, i=1,2,...,m, are guaranteed by continuity to be bounded for p sufficiently large; however, this condition may hold for modest values of p, even p = 0. By the sufficiency conditions, $\{|\lambda_i^*|\}$, i=1,2,...,m are bounded, and since $\lim_{p\to\infty} \rho c_i(x^*(\rho)) = -\lambda_i^*$, it follows that $|c_i(x^*(\bar\rho))|$ is of order $\frac{1}{\bar\rho}$ for all i. Hence $||G\Delta x||$ and $||\sum_{i=1}^{m} \bar\rho c_i(x^*(\bar\rho))G_i\Delta x||$ are of the order of $||\Delta x||$, i.e., $\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)$

Because of the non-tangential approach of $x^*(p)$ to x^* (see Section 2.2.3), $||\bar\rho AA^T\Delta x||$ will be of order $\left(\frac{\bar\rho}{\rho}\right)$ as will $||\bar\rho\Delta c||$. The following relationship thus holds at $x^*(\rho)$:

$$\bar\rho AA^T Ax = -g - \bar\rho Ac + O\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)$$

Because $A(x^*)$ has full rank, $A(x^*(\rho))$ is assured to be of full rank for ρ large enough, where, again, "large enough" need not imply a very large value of p. If we substitute for g from (5.2.1) the expression $(-\rho Ac)$, and cancel the full-rank matrix A, the result is:

$$A^T\Delta x = -\left(1 - \frac{\rho}{\bar\rho}\right)c + O\left(\frac{1}{\bar\rho}\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)\right) \qquad (5.2.2)$$

Exactly the same result can be derived from an alternative viewpoint. Since $\lim_{p\to\infty} \rho c(x^*(p)) = -\lambda^*$, and $\lambda_i^* \neq 0$ for any i, the estimate of the Lagrange multipliers at $x^*(p)$, given by:

$$\lambda(x^*(\rho)) = -\rho c(x^*(\rho)),$$

is bounded away from zero for ρ large enough, and is accurate to within order $\left(\frac{1}{\rho}\right)$ by the properties of the trajectory of minima of $P(x,\rho)$. If the requirement is imposed that the Lagrange multiplier estimates at $x^*(p)$ and $x^*(\bar\rho)$, > p, agree to order $||\Delta x||$, i.e., $\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right)$, the result is:

$$\rho c(x^*(p)) = \bar\rho c(x^*(\rho) + Ax) + O\left(\frac{1}{\rho} - \frac{1}{\bar\rho}\right). \qquad (5.2.3)$$

The expansion of c about $x^*(\rho)$ is given by:

$$c(x^*(\rho) + \Delta x) = c(x^*(\rho)) + A^T\Delta x + O(||\Delta x||^2) \ . \qquad (5.2.4)$$

Because of the non-tangential approach to x^*, $||A^T\Delta x||$ will be of order $\left(\frac{1}{\rho} - \frac{1}{\bar{\rho}}\right)$. Substituting the expansion (5.2.4) in (5.2.3) gives:

$$\rho c = \bar{\rho} c + \bar{\rho} A^T x + O\left(\frac{1}{\rho} - \frac{1}{\bar{\rho}}\right) ,$$

which is identical'to the previous result (5.2.2).

This restriction of the portion of Ax in the range of $A(x^*(\rho))$ exists because a change in the penalty parameter induces a specified first-order variation in the constraint values as x^* is approached along the trajectory of the quadratic penalty function. An interesting property of **Ax,** indicated by the derived relationship, is that the step Δx is less than the estimated first-order step to a zero of the constraint vector, because the factor multiplying $(-c)$ on the right-hand side of (5.2.2) is less. than unity. This characteristic is a refinement of the well-known property that the signs of $\{c_i(x^*(\rho))\}$ remain constant for p sufficiently large.

Throughout this section, it has been assumed that ρ is "sufficiently large'' for the various assumptions to hold. However, it should be emphasized that these restrictions do not mean that $x^*(\rho)$ is in a close neighborhood of x^*. The result (5.2.2) may hold even for modest values of ρ, provided that $\bar{\rho}$ is near enough to ρ.

Inequality Constraints

The relationships derived for the equality case will also hold under suitable assumptions for the quadratic penalty function applied to the inequality constrained problem, P2:

$$P(x,\rho) = F(x) + \frac{\rho}{2} \sum^{\ell} (\min(0, c_i(x)))^2$$

As seen in Chapter 2, for sufficientiy large ρ the set of violated constraints at $x^*(\rho)$ is identical to the set of active constraints at x^*. It will be assumed that m ccnstraints are violated at $x^*(\rho)$, and the vector $\hat{c}$ is used to denote the vector of violated constraints, with a similar conventicn for $\hat{A}$.

If the set of violated constraints at $x^*(\bar{\rho})$ is the same as the violated set at $x^*(\rho)$, the derivation previously given will apply to the inequality problem, and the result (5.2.2) can be written:

$$\text{ATAx} = -\left(1 - \frac{\rho}{\bar{\rho}}\right)\hat{c} + O\left(\frac{1}{\bar{\rho}}\left(\frac{1}{\rho} - \frac{1}{\bar{\rho}}\right)\right) .$$

5.3 Further Properties of the Logarithmic Barrier Functicn

In this section, we shall be concerned with solution of the inequality constrained problem, P2, by using the logarithmic barrier function first described in Chapter 2:

$$B(x,r) = F(x) - r\sum_{i=1}^{\ell} \ln(c_i(x)).$$

The notation used will be identical to that defined in Chapter 2, so that the set of active constraints is represented as follows. The m constraints active at x^* are grouped in the vector $\hat{c}$, whose elements are numbered $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_m$, where "$\hat{c}_i$" denotes the ith active constraint; a similar convention applies to the n by m matrix $\hat{A}$ whose columns are the gradients of the active constraints, and to $\hat{G}_i$, which denotes the Hessian of the ith active constraint. The m Lagrange multipliers $\{\lambda_i^*\}$ are numbered to correspond with the components of $\hat{c}$.

It will be assumed that: the sufficient conditions given in Section 1.2 for a minimum of P2 hold at x^*, $\hat{A}(x^*)$ has full rank, $\lambda_i^* \neq 0$ for any i, $||G||$ and $||G_i||$, $i=1,2,\ldots,\ell$, are bounded at x^*, and $\lim_{r \to 0} x^*(r) = x^*$.

At the minimum of B(x,r),

$$\nabla B(x^*(r),r) = g-rA \begin{bmatrix} \frac{1}{c_1} \\ \vdots \\ \vdots \\ \frac{1}{c_\ell} \end{bmatrix} \equiv g-rAd = 0, \qquad (5.3.1)$$

where the function d is defined as the vector $\left(\frac{1}{c_1}, \ldots, \frac{1}{c_\ell}\right)^T$

The notation $\hat{d}$ will refer to the vector $\left(\frac{1}{-}, \frac{1}{-}, \quad , \quad {}_{m}\right)^T$, which includes only the active constraints.

Consider the perturbation Ax = $x^*(\bar{r}) - x^*(r)$, where $\bar{r} < r$ and $(r-\bar{r})$ is small relative to r and $\bar{r}$. By definition of $x^*(\bar{r})$:

$$g(x^*(\bar{r})) = \bar{r}A(x^*(\bar{r}))\ d(x^*(\bar{r})).$$

Expanding g and A in their Taylor series about $x^*(r)$ yields:

$$g + G\Delta x = \bar{r}Ad(x^*(\bar{r})) + \sum \bar{r}d_i(x^*(\bar{r}))G_i\Delta x + O(||\Delta x||^{-}).$$

We now estimate the size of each term. Because of the properties of the trajectory of B(x,r) (see Section 2.4.3), $||\Delta x|| = O(r-\bar{r})$. The quantities $||G||$ and $||G_i||$ are bounded at x^*, so that $||G(x^*(r))||$ and $||G_i(x^*(r))||$, $i=1,2,\ldots,\ell$, are guaranteed by continuity to be bounded for r sufficiently small; however, these quantities may be boundea for any value of r, so that "sufficiently small'' r may not imply that r is close to zero. For $\bar{r}$ small enough, the components of d corresponding to inactive constraints are strictly bounded, and thus the term $\bar{r}\ d_i(x^*(\bar{r}))$ is of order $\bar{r}$ if c_i is inactive at x^*. By the sufficiency conditions for a minimum, $||\lambda^*||$ is bounded; since $\lim_{r \to 0} \frac{r}{\hat{c}_i(x^*(r))}$ – $\lim_{r \to 0} r\ \hat{d}_i(x^*(r)) = \lambda^*_i$, it follows that for $\bar{r}$ small enough the component

$\bar{r}\, d_i(x^*(\bar{r}))$ corresponding to an active ccnstraint is bounded. Therefore, all elements of the sum $\sum_{i=1}^{\ell} \bar{r}\, d_1(x^*(\bar{r}))\, G_1 \Delta x$ are of order at most **Ax,** i.e., $\mathcal{O}(r-\bar{r})$. Grouping together all terms of order $(r-\bar{r})$ or less, the result is:

$$g = \bar{r} A d(x^*(\bar{r})) + \mathcal{O}(r-\bar{r}).$$

At $x^*(r)$, $g = rAd(x^*(r))$; substituting this value for g gives:

$$rAd(x^*(r)) = \bar{r}Ad(x^*(\bar{r})) + \mathcal{O}(r-\bar{r}), \text{ or}$$

$$rA \begin{bmatrix} \frac{1}{c_1(x^*(r))} \\ \frac{1}{c_2(x^*(r))} \\ \vdots \\ \frac{1}{c_\ell(x^*(r))} \end{bmatrix} = \bar{r}A \begin{bmatrix} \frac{1}{c_1(x^*(\bar{r}))} \\ \frac{1}{c_2(x^*(\bar{r}))} \\ \vdots \\ \frac{1}{c_\ell(x^*(\bar{r}))} \end{bmatrix} + \mathcal{O}(r-\bar{r}), \qquad (5.3$$

where A is evaluated at $x^*(r)$.

The relationship (5.3.2) may hold anywhere along the trajectory if $\bar{r}$ is sufficiently close to r, and the assumptions inherent in the hounding procedure are satisfied. However, because $A(x^*(r))$ is not necessarily of full rank, it is not possible in general to draw any meaningful conclusion about the variation in the constraint values. To refine (5.3.2) to correspond to the result in the penal-ty case, it is necessary to assume that the value of r is snfficiently small so that an $\mathcal{O}(r)$ term is negligible; then the active and inactive constraints may be considered separately. Since the inactive constraints are bounded away from zero, the components $\left\{\frac{r}{c_i(x^*(r))}\right\}, \left\{\frac{\bar{r}}{c_i(x^*(\bar{r}))}\right\}$ corresponding to inactive constraints may be included in an order r term, leaving a relationship that holds for the active constraints:

$$r\hat A\begin{bmatrix}\frac{1}{\hat c_1(x^*(r))}\\ \frac{1}{\hat c_2(x^*(r))}\\ \vdots\\ \frac{1}{\hat c_m(x^*(r))}\end{bmatrix} = \bar r\hat A\begin{bmatrix}\frac{1}{\hat c_1(x^*(\bar r))}\\ \frac{1}{\hat c_2(x^*(\bar r))}\\ \vdots\\ \frac{1}{\hat c_m(x^*(\bar r))}\end{bmatrix} + O(r) .$$

All elements in the denominators are bounded away from zero for $r,\bar r \neq 0$, and hence the ith row of each vector can be multiplied by the factor $\frac{\hat c_i(x^*(r))}{r}\,\hat c_i(x^*(\bar r))$, which is of order $\bar r$, yielding:

$$\hat A\hat c(x^*(\bar r)) = \frac{\bar r}{r}\,\hat A\hat c(x^*(r)) + O(r\cdot\bar r) .$$

Expanding $\hat c$ in a Taylor series about $x^*(r)$ gives:

$$\hat A\hat c + \hat A\hat A^T\Delta x = \frac{\bar r}{r}\,\hat A\hat c + O(r\cdot\bar r) .$$

Because of the non-tangential approach of $x^*(r)$ to x^*, the term $\hat A\hat A^T$Ax is of order Ax, i.e., $(r-\bar r)$. Since $\hat A(x^*)$ has full rank, $\hat A(x^*(r))$ is guaranteed by continuity to be of full rank for sufficiently small r. Cancelling the full-rank matrix $\hat A$ and re-arranging, we obtain the desired characterization of Ax:

$$\hat A^T\Delta x = -\left(1-\frac{\bar r}{r}\right)\hat c + O(r\cdot\bar r), \qquad (5.3.3)$$

which is similar to that given for the penalty [illegible]

the previcus section.

An alternative derivation can be given by requiring the Lagrange multiplier estimates for active constraints at $x^*(r)$ and $x^*(\bar r)$ to agree to within order r. At $x^*(r)$, the ith multiplier estimate is $\frac{r}{\hat c_i(x^*(r))}$, so that the requirement is

$$\frac{r}{\hat c_i(x^*(r))} = \frac{\bar r}{\hat c_i(x^*(\bar r))}$$

Multiplying each side by $\frac{\hat{c}_i(x^*(r))\,\hat{c}_i(x^*(\bar{r}))}{r}$ and expanding $\hat{c}_i$ about $x^*(r)$, we again obtain (5.3.3).

As with the penalty function, it is possible to characterize a step along the trajectory of approach because the first-order variation of the active constraints is controlled by the barrier parameter. A key restriction in the barrier function case is that the inactive constraints cannot be ignored except for small values of the barrier parameter, since their influence is always present in a barrier function algorithm.

5.4 A Penalty Trajectory Algorithm

5.4.1 General Description

The penalty trajectory algorithms to be discussed will be concerned initially with the equality constrained problem, P1, and extended to deal with inequality constraints in Section 5.4.4.

In a penalty trajectory algorithm, the sequence of iterates is constructed so that its path approaches the trajectory of the quadratic penalty function method converging to x^*. A penalty trajectory algorithm retains in every step at least one penalty paraneter with the usual significance, to serve as a measure of progress toward the solution and ultimately to be increased to infinity to enforce convergence. However, the penalty parameter is monitored and possibly altered at every iteration, and does not remain fixed until an unconstrained minimization of a penalty function is completed. Its influence in a trajectory algorithm tends to generate points close to the penalty trajectory, but does not cause increasingly poor conditioning of the linear systems to be solved.

At each iteration of a penalty trajectory algorithm, we wish to compute a step toward the trajectory of $x^*(\rho)$. Because the constraint values at $x^*(\rho)$ for sufficiently large ρ are related in a known way to the Lagrange multipliers at x^*, and the approach of $x^*(\rho)$ to x^* is not tangential to any constraint, the portion of a move toward $x^*(\rho)$ that causes a first-order change in the constraints tends to satisfy certain conditions. The key characteristic of a penalty trajectory algorithm is that the portion of the search direction in the range of A at the current point, which generates a first-order change in the constraints, is specified by a system of linear equality constraints derived from the properties of the quadratic penalty function trajectory. This derivaticn contrasts with the linear equality constraints of the Lagracgian-type inethods discussed in Chapter 4, which are based on conditions holding only at x^*. As the penalty parameter in a trajectory algorithm approaches infinity, however, the portion of the search directior, in the range of A becomes arbitrarily close to the analogous portion of the step taken by a linearly constrained Lagrangian method.

Much of the discussion in this chapter will be concerned with quadratic programming problems; the following I s a brief review of the necessary background (see Gill and Murray, 1974b, for further details).

Consider the quadratic programming problem:

$$\underset{p}{\text{minimize}}\ \tfrac{1}{2}\, p^T S p + p^T d$$

$$\text{subject to } A^T p = u,$$

where A is n by m, of rank $m \leq$ ri. The solution p^* can be written as the sum of two orthogonal components, i.e., $p^* = Q_1 p_R + Z p_N$. Q_1 is an n by m matrix whose columns form an orthogonal basis for the column space of A, and Z is an n by (n-m) matrix whose columns form an orthogonal basis for the null space of A.

A particular representation of Q_1 and Z car, be obtained by partitioning the orthogonal matrix Q in the factorization of A^T (see Gill, Golub, Murray, an3 Saunders, 1974), defined by:

A^T (see Gill, Golub, Murray, and Saunders, 1974), defined by:

$$A^T = [L \vdots 0]\ Q = [L \vdots 0] \begin{bmatrix} Q_1^T \\ \hdashline \end{bmatrix} \begin{matrix} \} \ m \\ \} \end{matrix} .$$

When A^T is of full rank, L is a non-singular m by m lower triangular matrix satisfying:

$$A^T Q_1 = L .$$

This factorization may be computed in various ways, in particular by application of Householder transformations or plane rotations (see Chapter 6 for additional discussion).

The vector $Q_1 p_R$ will be termed the component of p^* in the range of A, and ZpN will be referred to as the null-space component of p^* Because the columns of Q_1 and Z are orthogona , it follows that:

$$\|p^*\|^2 = \|Q_1 p_R\|^2 + \|Z p_N\|^2 = \|p_R\|^2 + \|p_N\|^2 .$$

The vector pR is uniquely determined by the equality constraints, since:

$$A^T p = A^T (Q_1 p_R + Z p_N) = A^T Q_1 p_R = L p_R = u, \text{ and}$$

L is by definition non-singular.

The vector p_N is defined by the minimization of the quadratic function in the null space of A, and is the solution of:

$$Z^T S Z p_N = -Z^T d - Z^T S Q_1 p_R .$$

For p^* to be a strict minimum, the matrix $Z^T SZ$ must be positive definite.

It should be noted that any portion of' the victor d in the range of A is annihilated by the application of Z^T, so that the solution p^* is not affected if terms of the form Av fer some v are included in d.

5.4.2. Specification of the Search Direction arid Step Length

Murray's Original Trajectory Algorithm

From the results of Section 5.2, we see that an approximation, p, to the step Ax from $x^*(\rho)$ to $x^*(\bar{\rho})$, $\bar{\rho} > \rho$, will under certain conditions satisfy with high accuracy the linear equality constraints:

$$\text{LC1 :} \qquad A^T p = -(1-\frac{\rho}{\bar{\rho}})c,$$

where A and c are evaluated at $x^*(\rho)$. Eased on the assumption that the current value of x is close to $x^*(\rho)$, the original penalty trajectory algorithm of Murray (1969a,b) solves the following quadrati; programming problem to obtain the search direction at each step:

$$\text{QP1:} \qquad \text{minimize } \tfrac{1}{2} p^T S p + p^T g$$

$$\text{subject to } A^T p = -(1-\frac{\rho}{\bar{\rho}})c \ .$$

In QP1, the vector g is as usual the gradient of F(x), and ρ is the current estimate of the penalty parameter; $\bar{\rho}$ is the penalty parameter such that x+p is an estimate of $x^*(\bar{\rho})$.

Murray discussed three possible choices for the function to be approximated by the quadratic objective function in QP1 : F(x), the penalty function $P(x,\rho)$, or an approximate Lagrangien function $L(x,\lambda) = F-\lambda^T c$ for some vector λ of estimated Lagrange multipliers. Regardless of which choice is made, only the gradient of F(x) needs to be included in the linear term of the quadratic function; because the gradients of both $P(x,\rho)$ ana $L(x,\lambda)$ consist of g plus a vector of the form Av for some v, this latter term is annihilated by Z^T when computing p_N (see the discussion of quadratic programs in Section 5.4.1). The matrix S is the exact or approsimate Hessian of one of the three choices.

Modified Trajectory Algorithm

The 'linear constraints LC1 are based on the assumption that the current point is the minimum of the quadratic penalty function with penalty parameter ρ, and thus a direction satisfying LC1 is always a local direction of descent for c_i^2 for every constraint. This property is unnecessarily restrictive for a general algorithm, since at times it may be desirable to increase infeasibility for some, or even all, of the constraints. A set of linear equality constraints that characterize a step toward the penalty function trajectory in a more general way can be derived if the current point is "close" to the trajectory, and an estimate of the Lagrange multiplier vector is available.

The method for computing the estimate of the Lagrange multipliers will not be specified here, since there are various possibilities, depending on the information available. The only assumptions are that the method is always defined, and is consistent in a neighborhood of x^*, i.e., $\lambda \to \lambda^*$ as $x \to x^*$.

The condition sought on the search direction, p, is that x+p be a good approximation to $x^*(\bar{\rho})$ for some $\bar{\rho}$. At $x^*(\bar{\rho})$, the vector $-\bar{\rho}c(x^*(\bar{\rho}))$ is an estimate of the Lagrange multipliers at x^*, with accuracy related to $\bar{\rho}$. Hence, using the current multiplier approximation, λ, we require that:

$$-\bar{\rho}c_i(x+p) \doteq \lambda_i, \quad i=1,2,...,m.$$

Because of the non-tangential approach to x^*, it is reasonable to approximate $c_i(x+p)$ by only the first-order term of the Taylor series expansion about x. Ignoring all but first-order terms, the result, is:

$$-\bar\rho c_i = \bar\rho a_i^T p \equiv \lambda_i; \text{ so that}$$

$$a_i^T p = -c_i - \frac{\lambda_i}{\bar\rho}, \quad i=1,2,\ldots,m, \text{ or}$$

Lc2: $\quad A^T p = -c - \frac{1}{\bar\rho} \lambda,$

where A and c are evaluated at the current point. This specification of the portion of p in the range of the current A is a first-order prediction that each constraint value at the updated point will satisfy the eppropriate relationship with the corresponding multiplier estimate and the penalty parameter. Algorithms with similar motivation, based on Murray's method, nave been described by Biggs (1972, 1974).

It should be noted that if the current point is sufficiently close to $x^*(\rho)$, the difference in the formulations LC1 and LC2 is negligible. The iirst-order estimate of λ at $x^*(\rho)$ is $-\rho c$; substituting this expression for λ in LC2, the right-hand side becomes $(-c + \frac{\rho}{\bar\rho} c)$, as in LC1. Furthermore, as $\bar\rho$ goes to infinity, the relation LC2 approaches the specification of a first-order step to a zero of the constraints, $A^T p = -c$.

The linear equality constraints to be satisfied by the search direction in all the penalty trajectory algorithms to be considered are given by LC2. This characterization uniquely determines the vector $Q_1 p_R$, since p_R satisfies:

$$A^T Q_1 p_R = L p_R = -c - \frac{\lambda}{\ }$$

where the lower triangular matrix L is as described in Section 5.4.1, and is by definition non-singular.

The method for computing the null-space component of the search direction follows the idea originally given by Murray; p_N is chosen to minimize a quadratic approximation to some function, which is one of: $F(x)$, $P(x,\rho)$, or $L(x,\lambda)$ for an estimated multiplier vector λ.

The linear term of the quadratic function is always g^Tp, since the other terms in the gradients of the given choices are of the form Av for some v, and do not affect the calculation of p_N. The Hessian matrix of the quadratic function is the exact Hessian or an approximation 'io the Hessian of one of the three choices.

The usual strategy for calculating p_N is that p_N minimizes the particular quadratic approximation in the null space of A, given that the move Q_1p_R has been made from the original point in order to,satisfy the linear constraints. Tnis logic makes the search direction the sclution of the quadratic programming problem:

QP2: minimize $\frac{1}{2}\,p^TSp + p^Tg$

subject to $A^Tp = -c - \hat{c}$.

The search direction is then. given by $Q_1p_R + Zp_N$, where Q_1p_R is determined by the linear equalities and p_N solves the equations:

E2: $$Z^TSZp_N = -Z^Tg - Z^TSQ_1p_R \;.$$

The vector p_N can alternatively be chosen to minimize the quadratic function independently of Q_1p_R, by assuming that the only move from the current point will be in the null space of A. With this formulation, p_N solves the following set of linear equations:

E1: $$Z^TSZp_N = -Z^Tg \;.$$

This definition of pN might be used in the context of the general approach of computing the search direction a:: a linear combination of two orthogonal components. The direction that solves QP2 is not necessarily a direction of descent for the quadratic functicn to be minimized, since the move required to satisfy the linear constraints may increase the function from its value at the current point. The linear constraints to be satisfied by the search direction are based on the combined assumptions of closeness to the penalty trajectory and reasonable multiplier estimates. When these conditions are uncertain, it seems logical that the null-space portion of the search direction should be independent of the linear constraints, so that an appropriate linear combination of Q_1p_R and Zp_N can be a descent direction for the quadratic function.

After the search direction has been determined, a linear search is carried out to obtain a step length $\alpha > 0$ such that $x + \alpha p$ is an approximate minimum of some function. The two choices for this function are: $P(x,\hat{\rho})$ for some $\hat{\rho}$, or an augmented Lagrangian function, $L(x,\lambda,\tilde{\rho}) = F - \lambda^T c + \frac{\tilde{\rho}}{2} c^T c$, for some $\tilde{\rho}$. The first choice is the more usual and straightforward; it will guarantee a reduction at every iteration in a penalty function, thereby providing a standard of progress toward the solution. The second function is chosen by reasoning similar to that given for the Lagrangian-type methods described in Chapter 4.

Summary

The essential elements of a penalty trajectory algorith are:

(1) maintenance of a continuously monitored penalty parameter to regulate convergence as in a standard penalty function method;

(2) calculation of the search direction, p, as

$p = Q_1p_R + Zp_N$, such that:

(a) Q_1p_R is computed so that p satisfies a system of linear equality constraints of the form $A^Tp = d$, where the vector d depends on the controlling penalty parameter and the current estimate of the Lagrange multipliers; and

(b) Zp_N minimizes a quadratic approximation to some function;

(3) a linear search at every iteration to assure progress toward x^* by a guaranteed decrease in some suitable function.

The above description of a penalty trajectory algorithm is quite general, and there is room for variation in details of strategy and implementation (see discussion in Chapter 6). However, all methods of this class possess the same distinctive motivation, based on the quadratic penalty function.

5.4.3 Properties of the Penalty Trajectory Algorithm

Properties of p_R

An important property of points on a penalty function trajectory is that for a sufficiently large value of the penalty parameter, each ccnstraint value is of sign opposite to the corresponding Lagrange multiplier at x^*; when this condition holds at a point, the point is said to be on the "penalty side" of the constraint. If the current point is on the penalty side of all the ccnstraints, $\bar{\rho}$ is sufficiently large, and all multiplier estimates have the correct sign, the step induced by the linear constraints

LC2 will tend to decrease the constraint penalty for each constraint while remaining on the penalty side of all constraints, since it is 'a descent direction for c_i^2, $i=1,2,\ldots,m$, and an underestimate of the first-order step to a zero of all the constraints.

If a constraint value is close to zero at a point, the point is said to be "near" the constraint; if a constraint is zero, the point is said to be "on" the constraint; if a step causes an increase in infeasibility for a particular constraint, the move is said to be "away from" the constraint. It is desirable for an algorithm to be able to generate search directions that move away from certain constraints under appropriate circumstances - for example, if a point happens to be near a subset of the Constraints, yet far from x^*. If a method is committed to produce directions that yield a first-order decrease in violation for every constraint at each step, the method may become "trapped" into a sequence of inefficient small moves close to the constraints. An advantage of the formulation LC2 is that the search direction is not necessarily a descent direction for every constraint. If a constraint value is "too small" for the given penalty parameter, i.e., for some i $\bar{\rho}|c_i|$ is much smaller than $|\lambda_i|$, and λi has the correct sign, the step given by LC2 will tend to move away from the $\underline{i}$th constraint in order to satisfy the conditions on the penalty trajectcry.

In rare cases, normally only in the neighborhood or" the starting point, it can happen that a constraint is zero far from x^*, or that a constraint and the corresponding multiplier estimate agree in sign. In this event, either the current point is not on the penalty side of the given constraint, or the multiplier estimate has the wrong sign. The move given by LC2 in such a case is a prediction that the next point will Lie on tile other side of the constraint, since the underlying assumption is that the sign of the multiplier estimate is correct. The best way to specify $a_i^T p$ for this situation is not known (see Chapter 6 for details of current strategy); the approach selected depends on which of the two alternatives (x is not on the penalty side of c_i, or λ_i has the wrong sign) is considered more likely at the current point. In either case, it is always possible to choose the search direction so that the desired descent properties with respect to the original problem are retained.

Calculation of p_N

Any term included in the matrix S used to compute p_N which is a multiple of ADA^T, for symmetric non-singular D, is annihilated by application of Z^T, and does not affect the resulting p_N (see the discussion of quadratic programming in Section 5.4.1). Thus, if S is taken to be the Hessian matrix of the penalty function $P(x,\rho)$, the portion of S that causes ill-conditioning for the usual penalty function method, i.e., ρAA^T, has no effect on the search direction in the trajectory algorithm.

To obtain the local convergence rates of linearly constrained Lagrangian methods close to x^*, the matrix S should be chosen so that Z^TSZ is a good approximation to Z^TWZ, where $W = G - \sum_{i=1}^{m} \lambda_i^* G_i$. If the current point is very close to $x^*(\rho)$ for sufficiently large ρ, the choice of S as either the Hessian of $P(x,\rho)$ or $L(x,\lambda)$ will serve to make Z^TSZ an approximation to Z^TWZ, because the portion of $V^2P(x,\rho)$ that remains after application of $Z^\perp$ is given by $G + \sum_{i=1}^{m} \rho c_i G_i$, and $(-\rho c_i)$ is an approximation to λ_i at $x^*(\rho)$. The formulation of S involving λ is generally preferred close to x^*, since its independence of the penalty parameter should achieve continuity of Z^TSZ in a neighborhood of x^*.

The formulation given for computing p_N does not require that S be positive definite, or even non-singular. However, the specification that p_N [illegible] matrix is S in the null space of A, implies that Z^TSZ should be positive definite. At x^*, Z^TWZ is positive definite, and if Z^TSZ is a good approximation to Z^TWZ, then Z^TSZ should be positive definite in a neighborhood of x^*. Nonetheless, to avoid complications at the inevitable times when this matrix is not positive definite, the linear system for p_N is solved with a modified Cholesky algorithm (see Gill and Murray, 1972a, for details). If the matrix Z^TSZ is sufficiently positive definite, it will be unaltered in solving the linear system for p_N; if it is not sufficiently positive definite, it is replaced by a neighboring matrix guaranteed to be positive definite. If Z^TSZ is not positive definite, it is imperative that the calculation of the null-space component be altered along these lines; otherwise, p_N is a step toward a saddle point, and may not even be a descent direction for the quadratic function which it is designed to minimize.

To avoid additional notation, p_N will usually be written in terms of the matrix Z^TSZ; but this representation should be interpreted to mean a matrix that is always positive definite, regardless of the properties of the unaltered matrix.

Descent Property

The search direction, p, generated by a penalty trajectory algorithm is guaranteed to be a descent direction for a penalty function or an augtiisnted Lagrangian function, given a suitable choice of the various penalty parameters. This property is significant because at each step of a penalty trajectory algorith a linear search is executed to find a positive step along the search direction to the approximate minimum of' one of these two choices.

The result will be demonstrated by using the linear equalities LC2 satisfied by **p**. For convenience, the penalty parameters of the penalty function and the augmented Lagrangian function will both be denoted by $\hat\rho$, since the required size of either parameter can be deduced from the proof'. The gradient of either the penalty or augmented Lagrangisn function can then be written as $g - A\hat\lambda + \hat\rho Ac$, where $\hat\lambda = 0$ far the penalty function, and $\hat\lambda$ is the current multiplier estimate for the augmented function.

The descent property will be verified if the scalar product of p with the appropriate gradient can be guaranteed to be negative. The scalar product is:

$$(g - A\hat\lambda + \hat\rho Ac)^T p =$$

$$(g - A\hat\lambda)^T p + \hat\rho c^T A^T p \ .$$

Because $A^T \mathbf{p} = -c - \frac{\lambda}{\bar\rho}$, the above expression becomes:

$$(g - A\hat\lambda)^T p + \hat\rho c^T(-c - \frac{\lambda}{\bar\rho}) \ . \qquad (5.4.1j$$

For $\bar\rho$ sufficiently large, the first term in (5.4.1) is bounded in magnitude by quantities independent of $\bar\rho$, by construction of p. If $|c_i| \neq 0$, $i=1,2,\ldots,m$, then for $\bar\rho$ sufficiently large the sign of each component of the vector $(-c - \frac{A}{\bar\rho})$ will be the sign of the appropriate component of $-c$, so that the second term in (5.4.1) will be negative. Therefore, if $\hat{\rho}$ is large enough, the sign of the second term will dominate the scalar product, and p is indeed a descent direction. A similar argument can be made provided at least one component of c is non-zero.

If $||c|| = 0$, the strategy used by a penalty trajectory algorithm will compute a search direction that decreases the objective function, and hence the appropriate descent property will be satisfied for a penalty function, since $\nabla P(x,\rho) = g$ if $||c|| = 0$. For the augmented function with $||c|| = 0$, tne scalar product of p and the gradient becomes:

$$p^T(g - A\lambda) =$$
$$p^Tg - p^TA\lambda =$$
$$p^Tg - (-\frac{\lambda}{\bar{\rho}})\lambda =$$
$$p^Tg + \frac{\lambda^T\lambda}{\bar{\rho}} .$$

If $p^Tg < 0$, and $|\quad||$ is bounded, then for sufficientiy large, the term $\frac{\lambda^T\lambda}{\bar{\rho}}$ will become negligible, and p is therefore a descent direction for the augmented Lagrangian function.

The proof' given for the case when $||c|| \neq 0$ rests OD two conditions: p can be made a descent direction for c^Tc by choosing $\bar{\rho}$ sufficiently largo in the vector on the right-hand side of the linear constraints LC2; and the gradient of either a penalty function or an augmented Lagrangian function can be dominated at such a point by the gradient of the penalty term $\frac{\rho}{2}c^Tc$ for sufficiently large $\hat{\rho}$. In fact, the descent property would hold for any direction that decrease; c^Tc, and hence the result is not of great usefulness. It merely assures that the linear search in a trajectory algorithm can always be successfully executed.

Local Convergence Properties

The local convergence properties of a penalty trajectory algorithm can be made essentially the same as those of a linearly constrained Lagrangian algorithm with a quadratic programming subproblem, if in the linear constraints LC2 is sufficiently large.

Let p_L denote the solution of the quadratic programming problem posed in a linearly constrained Lagrangian method:

$$\min \tfrac{1}{2} p^T S p + g^T p$$

$$\text{subject to } A^T p = -c,$$

where S is an approximation to W, the Hessian of the Lagrangian function. A method that takes a unit step along the search direction p_L has been shown to converge locally to x^* at a secsnd-order rate if the exact Hessians of F and $\{c_i\}$ are used in S, so that $S = G - \sum_{i=1}^{m} \lambda_i G_i$ (See Section 4.8.5); other convergence results for such methods have been given (see Robinson, 1973), depending on the approximations used in obtaining S.

If the matrix S used in the quadratic programing problem QP2 is given by $G - \sum_{i=1}^{m} \lambda_i G_i$ for the current estimate λ, or an approximation thereto, the search direction of the trajectory algorithm can be made arbitrarily close to p_L for the same S by choosing ρ sufficiently large. The perturbation from $(-c)$ on the right-hand side

away from singularity, the vector p_R that satisfies the equalities LC2 thus differs from the corresponding portion of p_L by a perturbation of order $(\frac{1}{\rho})$. If the identical matrix S is used in both formulations, the nuil-space portion of the trajectory search direction will similarly be perturbed from the null-space portion of pL by order $(\frac{1}{\tilde{\rho}})$, since the matrix Z^TSZ used to compute N is guaranteed to be strictly positive definite. Therefore, a penalty trajectory algorithm with the quadratic program search direction can achieve the local convergence rate of an analogous linearly constrained Lagrangian method if $\tilde{\rho}$ is large enough so that the order $(\frac{1}{\tilde{\rho}})$ perturbstion is insignificant with respect to the terms that determine the rate of convergence. Further comments on the local convergence of the trajectory algcrithm are given in Section 5.6.

When the vector p_N is computed to satisfy the alternative formulation:

$$Z^TSZp_N = Z^Tg,$$

and S is given by the Hessian of a penalty function or an approximate Lagrangian function, the null-space component of the trajectory search direction, Zp_N, becomes an arbitrarily close approximation to the null-space component of the Newton step to the minimum of the penalty function $P(x,\tilde{\rho})$ for $\tilde{\rho}$ sufficiently large. Murray (1971) derived explicit expressions for the eigenvalues and eigenvectors of the Hessian matrix of a penalty function, and showed that the inverse of the Hessian matrix $\nabla^2P(x^*(\tilde{\rho}),\tilde{\rho})$ for $\tilde{\rho}$ large enough is closely approximated by:

$$(\nabla^2P(x^*(\tilde{\rho}),\tilde{\rho}))^{-1} \doteq \frac{1}{\tilde{\rho}} A(A^TA)^{-2}A^T + Z(Z^T(G + \sum_{i=1}^{m} \tilde{\rho}c_iG_i)Z)^{-1}Z^T, \text{ or}$$

$$\doteq \frac{1}{\tilde{\rho}} A(A^TA)^{-2}A^T + Z(Z^T(G - \sum_{i=1}^{m}\lambda_iG_i)Z)^{-1}Z^T, \qquad (5.4.2)$$

where λ is an estimate of the Lagrange multipliers. These relations hold because the Hessian of $P(x,\tilde{\rho})$ evaluated 31 $x^*(\tilde{\rho})$ has been shown (*see* Chapter 2) to split into two parts -- $G + \sum_{i=1}^{m}\tilde{\rho}c_iG_i$, which approaches the Hessian of the Lagrangian function as $\rho \rightarrow \infty$, and so becomes independent of the penalty parameter; and the matrix ρAA^T.

The expressions for $(\nabla^2 P(x^*(\tilde\rho),\tilde\rho))^{-1}$ allow a close approximation of the Newton step, $\tilde p$, to the minimum of $P(x,\tilde\rho)$, which is given by:

$$p = -(\nabla^2 P(x^*(\tilde\rho),\tilde\rho))^{-1}(g+\tilde\rho Ac) .$$

The vector g may be decomposed into components in the range and null space of A, so that $g = As + Zv$ for some s and v, and hence $Z^Tg = v$. Using the expression (5.4.2) for $(\nabla^2 P(x^*(\tilde\rho),\tilde\rho))^{-1}$ because its independence of $\tilde\rho$ is explicit, the definition of $\tilde p$ separates into two orthogonal. components:

$$\tilde p \doteq -(\frac{1}{\tilde\rho} A(A^TA)^{-2}A^T(As+\tilde\rho Ac)) - Z(Z^T(G-\sum_{i=1}^{m}\lambda_i G_i)Z)^{-1}v .$$

The null-space component of $\tilde p$ can, therefore, be written as $Z\tilde p_N \doteq -Z(Z^T(G-\sum_{i=1}^{m}\lambda_i G_i)Z)^{-1}v$, which is exactly equal to the expression for the null-space component of the trajectory search direction,

$$Zp_N = -Z(Z^TSZ)^{-1}v,$$

if S is chosen as indicated.

These results show that local convergence can be proved for a penalty trajectory algorithm. However, as seen in Chapter 4, local convergence can be proved under sufficiently strong assumptions for any of the Lagrangian algorithms, some of which are based on conditions holding only in a small neighborhood of x^*. Although of some importance, local convergence does not guarantee a robust or effective algorithm. The strength of the penalty trajectory algarithm does not lie in its local convergence to x^*, since locally any number of methods will be successful, but rather its derivation in terms of properties of the quadratic penalty function that do not critically depend on being in a neighborhood of x^*.

5.4.4 Extension to Inequality Constraints

A penalty trajectory algorithm for inequality constraints is based on the same motivation as the trajectory algorithm for equalities, where only violated constraints are included in the penalty term. In this discussion, the matrix $\hat{A}$ denotes the matrix of currently violated constraint gradients, with a similar convention for $\hat{c}$. The violate6 constraints are numbered from 1 to m in the vector $\hat{c}$, where the value of m may vary from iteration to iteration, and denotes the number of currently violated cccstraicts (assumed always $\leq n$). The estimated Lagrange multipliers correspccd only to violated constraints, and $\hat{Q}_1$ and $\hat{Z}$ have the usual properties with respect to $\hat{A}$.

Soecification cf the Search Direction

With inequality constraints, there is uncertainty as to which constraints are active at the solution, arid all algorithms for the inequality problem must make some determination, explicit or implicit, about the active set. It is cheracteristic of ttis trajectory of the quadratic penalty function that at a point sufficiently close to $x^*(\rho)$ for ρ large enough, the set of violated constraints is identical to the set of constraints active at x^*. Therefore, an algorithm based on the quadratic penalty function Is justified in using the set of currently violated constraints as a prediction of the active set, although it must be possible to modify this assumption at any stage of the algorithm, particularly near the starting point.

The relationship between the sets of violated and active constraints is reflected in the linear equalities at the end of Section 5.2 for the step along the penalty function trajectory, which are based only on the behavior of the currently violated constraints. As the estimates of the solution approach x^* along the penalty function trajectory, these linear equalities will include precisely the set of active constraints at x^*, and, as expected, specify the first-order variation in the active constraints only. it should be noted that the derivation of the linear constraints of the lineariy constrained Lagrangian methods for inequalities is also based on including only the active constraints.

Ey reasoning similar to the equality case, the trajeetcry search direction for inequality constraints should satisfy a set of equalities analogous to LC2:

LC2': $\hat{A}^T p = -c - \frac{\lambda}{\rho}$,

where λ is the current estimate of the Lagrange multipliers. The constraints LC2' provide a prediction of the violated set at the next step, based on the signs of the multiplier estimates, if ρ is sufficiently large. If sign (λ_i) = - sign $(\hat{c}_i)$, so that $\lambda_i > 0$, a step p that satisfies the <u>i</u>th constraint of LC2' gives a first-order prediction that the ith constraint will continue to be violated at the next point; if λ_i is negative, the satisfacticn of the <u>i</u>th constraint of LC2' amounts to a first-order prediction that the <u>i</u>th constraint will not be violated at the next point, since the value on the right-Land side is positive and exceeds $\hat{c}_i$ in magnitude. The initial step of the penalty trajectory algorithm for inequalities is thus computed to satisfy the linear constraints LC2', and the null-space component of the search direction is celculated as described for the equality case.

However, a more accurate prediction of the active set and a better search direction may be obtained through the following quadratic programming problem, where the linear constraints LC2' are generalized to a set of inequalities:

QP2': minimize $\frac{1}{2} p^T S p + p^T g$

subject to $\hat{A}^T p \geq - \hat{c} - \frac{\lambda}{\rho}$.

The initial trajectory search direction solves QP2' with all constraints taken as equalities. The Lagrange multipliers of QP2' for this solution provide a nigher-order estimate of the Lagrange multipliers of the original nonlinear problem, and so may be used to alter the predicticn of the active set, as follows: a negative multiplier of QP2' indicates that the quadratic objective function can be further decreased by deleting one of the active linear constraints. A modified search direction can thus be computed which solves QP2' with a reduced active set; with the new search direction, the first-order variation in the constraint predicted to be inactive is not restricted to satisfy the corresponding constraint as an equality·

At each iteration of the trajectory algorithm, the search direction could be computed by solving completely the quadratic programming problem €$€?',possibly deleting several constraints and reevaluating the multipliers . However, a simpler procedure, which has been successful In practice, is to delete only one active linear constraint per iteration, and accept the solution of the altered quadratic program as the search direction. The constraint to be deleted can be selected, if there are several possibilities, based on some consistent criterion (for example, if the constraints are well-scaled, choose the constraint with the most negative multiplier).

Calculation of Step Length

After computing the search direction, a linear search is executed, as in the equality case, to locate $\alpha > 0$ such that $x + \alpha p$ is the (approximate) minimum of a penalty function or an augmented Lagrangian function.

Descent Property

The search direction resulting fron any method where the solution satisfies the Linear inequalities of QP2' is a descent direction for a penalty function $P(x,\hat{\rho})$ or an augmented Lagrangian function $L(x,\lambda,\bar{\rho})$, given a suitable choice of the penalty parameters. The descent property is verified by examining the scalar product of the search direction, p, and the gradient of the penalty or augmeiited Lagrangian function, which may be written:

$$p^T(g - \hat{A}\hat{\lambda}) + \hat{\rho}\, p^T\hat{A}\hat{c} =$$

$$p^T(g - \hat{A}\hat{\lambda}) + \hat{\rho} \sum_{i=1}^{m} \hat{c}_i \hat{a}_i^T p , \tag{5.43}$$

where $\hat{\lambda} = 0$ for the penalty function and $\hat{\lambda} = \lambda$ for the augmented Lagrangian function, and where the "active" set included in the augmented Lagrangian function is assumed to be the set of violated constraints.

The linear inequalities of QP2' require that $\hat{a}_i^T p \geq -c_i - \frac{\lambda_i}{\bar{\rho}}$; it follows that for $\bar{\rho}$ sufficiently large, each component of $\hat{a}_i^T p$

must be positive. For such a value of $\bar{p}$, each element of the summation in the second term of (5.4.3) is negstive. The first term on the right-hand side of (5.4.3) is bounded in magnitude, and if $\hat{p}$ is sufficiently large: the negative sign of the second term must dominate the entire expression. Hence, the scalar product Is negative for suitable $\bar{\rho}$, $\hat{p}$, and p is a descent direction. As in the equality case, this result is implied by the local dominance of the penalty term $\frac{\hat{p}}{2}\hat{c}^T\hat{c}$ in either the penalty or augmented Lagrangian function.

5.5 A Barrier Trajectory Algorithm

5.5.1 General Description

A barrier trajectory algorithm generates a sequence of iterates whose path approaches the trajectory of a logarithmic barrier function method converging to x^*, while maintaining strict feasibility at every successive estimate of the solution. At least one barrier parameter is involved in every step of a 'carrier trajectory algorithm, with the usual role: its decrease controls the rete at which the strictly feasible iterates are allowed to approach x^*, which lies on the boundary of the feasible region. However, the barrier parameter in a trajectory algorithm is regulated and possibly altered at every iteration, rsther than remaining fixed until the unconstrained minimization of a particular barrier function is completed.

At every iteration, a barrier trajectory clgorithm computes a step toward the trajectory of $x^*(r)$. The active constraint values at $x^*(r)$ for sufficiently small r must satisfy a known relationship with the Lagrange multipliers at X%, and the approach of $x^*(r)$ to x^* is not tangential to any active constraint. Accordingly, the portion of the step that generates a first-order change In the active constraint values is subject to certain conditions, so that tne active constraints at the subsequent point tend to satisfy the appropriate relationship. The distinctive characteristic of a barrier trajectory algorithm is that the portion of the search direction in the range of the matrix $\hat{A}$ of active constraint gradients at the current point is specified by a set of linear equality constraints derived from the properties of the logarithmic barrier function trajectory.

5.5.2 Specification of the Search Direction and Step Length

In all discussion of the barrier trajectory algorithm: the matrix A will be taken to include the gradients of all the constraints of the problem; $\hat{A}$ will denote a full-rank matrix whose columns are a subset of constraint gradients corresponding to the constraints currently considered active, with a similar convention for c an2 $\hat{c}$. The active constraints are considered to be numbered from 1 to m in the vector $\hat{c}$, where the value of m may vary from iteration to iteration, and denotes the number cf constraints currently considered active (always $\leq n$). The estimated Lagrange multipliers are assumed to correspond only to active constraints. $\hat{Q}_\perp$ and $\hat{Z}$ have the usual properties with respect to $\hat{A}$.

in Section 3.3, it was shewn that an approximation, p, to the step Δx from $x^*(r)$ tc $x^*(\bar{r})$, $\bar{r} < r$, will, under certain conditions, satisfy with high accuracy the following linear equality constraints:

LC3: $$\hat{A}^T p = -(1 - \frac{\bar{r}}{r})\hat{c} ,$$

where $\hat{A}$ and $\hat{c}$ are evaluated at $x^*(r)$.

The linear constraints LC3 are based on the assumption that the current point is the minimum of the logarithmic barrier function with barrier parameter r, and a direction satisfying LC3 is always a local direction of descent for $\hat{c}_i^2$ for every "active" constraint. This property is unnecessarily restrictive for a general feasible algorithm, since it may be desirable at times to increase some constraints, or the current prediction of the active set may be incorrect. A set of linear constraints that characterize a step toward the barrier function trajectory in a more general way can be derived if the current point is "close" to the trajectory, and an estimate of the Lagrange multiplier vector is available. The condition desired for the search direction, p, is that x + p be a good approximation to $x^*(\bar{r})$. At $x^*(\bar{r})$, the vector $\left(\frac{\bar{r}}{\hat{c}_1}, \frac{\bar{r}}{\hat{c}_2}, \ldots, \frac{\bar{r}}{\hat{c}_m}\right)^T$ for the m active constraints, is an estimate of the Lagrange multipliers at x^*, with accuracy related to $\bar{r}$. We thus require that for each active constraint:

$$\frac{\bar{r}}{\hat{c}_i(x+p)} \doteq \lambda_i, \text{ or}$$

$$\bar{r} \doteq \lambda_i \hat{c}_i(x+p), \quad i=1,2,\ldots,m.$$

Because of the non-tangential approach to x^*, a linear approximation to $\hat{c}_i(x+p)$ will be accurate for $||p||$ sufficiently small. Including only the first-order terms in the Taylor series expansion about x, the resulting condition is:

$$\bar{r} = \lambda_i \hat{c}_i + \lambda_i \hat{a}_i^T p, \text{ so that}$$

$$\hat{a}_i^T p = -\hat{c}_i + \frac{\bar{r}}{\lambda_i}, \quad i=1,2,\ldots,m, \text{ or}$$

LC4: $$\hat{A}^T p = -\hat{c} + \bar{r} \begin{bmatrix} \frac{1}{\lambda_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_m} \end{bmatrix} .$$

The equalities LC4 satisfied by p are based on the relationship that should hold along the barrier trajectory between the active constraint values and the multiplier estimates. If the current point is close to x*(r) for sufficiently small r, the formulation LC4 is essentially the same as LC3, since at x*(r) a first-order multiplier estimate λ_i is given by $\&_i$

All the barrier trajectory algorithms to be considered are based on the equality constraints LC4, which uniquely determine the portion of the trajectory search direction in the range of $\hat{A}$. If $p = \hat{Q}_1 p_R + \hat{Z} p_N$, and p satisfies LC4, then:

$$\hat{A}^T p = \hat{A}^T \hat{Q}_1 p_R = \hat{L} p_R = -\hat{c} + \bar{r} \begin{bmatrix} \frac{1}{\lambda_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_m} \end{bmatrix} ,$$

so that p_R is computed by solving a system of linear equations, since the lower-triangular matrix $\hat{L}$ is non-singular by construction.

The method for computing the null-space component of the search direction is based on a similar idea to the penalty trajectory algorithm: p_N is chosen to minimize a quadratic approximation to some function, which is one of: F(x), B(x,r), or an approximate Lagrangian function $L(x,\lambda) = F-\lambda^T c$.

The linear term of the quadratic function is always taken as $g^T p$, where $g = \nabla F$, since the other terms in the gradients of the second and third choices are not relevant in computing p_N (see Section 5.4.1). The Hessian of the quadratic function is the exact Hessian or an approximation to the Hessian of one of the three choices.

The two possible strategies for computing p_N are similar to the two options in the penalty trajectory algorithm. However, in the barrier trajectory algorithm, the usual procedure is to choose p_N to minimize the quadratic function by moving in the null space of $\hat{A}$ from the current point, independently of' the move in the range of $\hat{A}$, i.e., p_N solves the system of equations:

E3: $$\hat{Z}^T S\hat{Z}p_N = -\hat{Z}^T g.$$

The alternative specification of p_N, used near x*, takes the search direction as the solution of the quadratic programming problem:

QP4: $$\text{minimize } \tfrac{1}{2} p^T Sp + p^T g$$

$$\text{subject to } \hat{A}^T p = -\hat{c} + \bar{r} \begin{bmatrix} \frac{1}{\lambda_1} \\ \vdots \\ \underline{1} \end{bmatrix},$$

so that p_N satisfies:

E4: $$\hat{Z}^T S\hat{Z}p_N = -\hat{Z}^T g - \hat{Z}^T S\hat{Q}_1 p_R$$

The reason for customarily choosing p_N to satisfy E3, to be explored more fully in Section 5.5.3, is the difference between the roles of the forcing parameter and the constraint values for penalty and barrier. functions. In the penalty case, increasing the penalty parameter at a non-boundary point makes the local behavior of the penalty term c^Tc dominate the penalty function. The linear coostraints satisfied by the search direction in a penalty trajectory algorithm are controlled by the penalty parameter so that the resulting direction can always be made a descent direction for c^Tc, and hence for the penalty function. The descent properties of the solution of the quadratic programming sub-problem for the penalty trajectory algorithm *thus* depend only on the special form of the linear constraints, LC2.

For a barrier function, on the other hand, decreasing the barrier parameter makes the influence of the constraints lese significant, and the behavior of the objective function tends to dominate locally. The linear constraints LC4, satisfied by the search direction in a barrier trajectory algorith, affect only the first-order change in the active constraints, and do not in general tend to make the search direction a descent direction for the objective function. In fact, the move generated by the solution of the quadratic programming problem QP4 may tend to increase the barrier function for any reasonable value of the barrier parameter; therefore, the formulation QP4 cannot be used at a general point, because the trajectory algorithm demands a monotonic decrease in a barrier function at every iteration. **It** will. be seen, however, that for points close to $x^*(r)$ for sufficiently small r, the quadratic program search direction can be guaranteed to be a descent direction for an appropriate barrier function.

After the search direction has been determined, a linear search is carried out to obtain a step length $\alpha > 0$ such that $x + \alpha p$ is an approximate minimum of the barrier function $B(x,\tilde{r})$ for some $\tilde{r}$.

The selection of a barrier function to be decreased at every step is designed tc maintain an appropriately controlled distance between the iterates and the constraint boundary. Although a method could be devised wherein the linear search attempted to find the approximate minimum of some cthes function, restricted to feasible points, this strategy could result in a point arbitrarily close to the boundary, but far enough from the solution to cause difficulties on subsequent iterations. Because a barrier function allows a continuously regulated distance to be maintained between the estimates and the boundary, and because an efficient linear search is available for use with the logarithmic barrier function (see Chapter 3), no other choice seems appropriate.

The essential elements of a barrier trajectory algorithm are:

(1) maintenance of at least one continuously monitored barrier parameter whose decrease controls the approach of iterates to the boundary of the feasible region;

(2) calculation of the search direction, **p**, as $p = Q_1 p_R + \hat{Z} p_N$, such that:

(a) $\hat{Q}_1 p_R$ is computed so that p satisfies a system of linear equality constraints of the form $\hat{A}^T p = d$, derived from the behavior of the logarithmic barrier function trajectory, where the vector d depends on the controlling barrier parameter and the current estimate of the Lagrange multipliers; and

(b) $\hat{Z} p_N$ minimizes a quadratic approximation to some function;

(3) a linear search with respect to E barrier function at each iteration to guarantee a measure of progress toward x^*.

The above description of a barrier trajectory algorithm is quite general, and details cf strategy cr implementation may vary (see Chapter 6). However, all the methods of this class are distinctively characterized by their derivation in terms of rhe behavior of the logarithmic barrier function.

5.5.3 Properties of the Barrier Trajectory Algorithm

Properties of $\hat{Q}_1 p_R$

Because a barrier algorithm is restricted to feasible points only, all constraint values are strictly positive, and therefore have the same sign as the Lagrange multipliers at x^*. If the correct active set has been chosen, $\bar{r}$ is sufficiently small, and all the estimated multipliers are positive, 2 searcn direction satisfying LC4 will tend to decrease all the active constraints, but stop short of the first-order step to the boundary. For $\bar{r}$ sufficiently small, the step satisfying LC4 is arbitrarily close tc the first-crder step to a zero of each active constraint, given by $\hat{A}^T p = -e$.

However, it should be noted that the step generated by LC4 is not necessarily a local descent direction for each active constraint. A serious weakness that can critically impair even a standard barrier function algorithm is that a subset of the constraints may become small far from the solution: because of local influences. From such a point, directions along which the barrier function is decreasing tend to cause negligible changes in the already small constraint values, and, since feasibility must be maintained, a large number of subsequent iterations may make essentially no progress toward x^*. An efficient feasible algorithm should be able tc move away from a constraint boundary; this property is more important than in the penalty function case because in a feasible algorithm there is no freedom to move to the other side of the constraint. if λ_i is positive, and an active constraint is "too small" for the value of $\bar{r}$, i.e., the estimate $\frac{\bar{r}}{c_i}$ is larger than λ_i, the step given by $LC4$ will tend to increase the ith constraint and move away from the boundary, as it should to achieve a non-tangential approach to x^*. Numerical experiments (see Chapter 6) have clearly demonstrated the value of this property of a barrier trajectory algorithm.

If the current estimate of a multiplier is negative, the assumptions upon which the linear equalities $LC4$ were based are not satisfied, and an alternative method should be used to compute the search direction. In such a case, either: (1) the given constraint has been incorrectly classified as active, or (2) although the constraint is active, the multiplier estimate has the wrong sign. Various strategies are possible to deal with this situation, but the procedure followed depends on which of the two alternatives is considered more likely.

If the negative multiplier estimate is taken as a signai that the constraint has been incorrectly classified as active: the particular constraint is deleted from the "active set'!, and new estimates of the multipliers and search direction are then computed for the diminished set of active constraints. This strategy is distinct from the strategy of the penalty trajectory algorithm for inequalities because here the constraint Is deleted from the "active" set before solving for the search direction. In. the penalty case for inequalities, if $\lambda_i < 0$, the step given by the usual relation for $\hat{a}_i^T p$ is a prediction that the ith constraint will not be violated at the next point, so that even the predicted move with the negative multiplier in essence deletes the ith constraint from the active set. In the barrier case, however, if $\lambda_i < 0$, the expression given for $\hat{a}_i^T p$ by LC4 predicts that the next point will be infeasible with respect to the ith constraint, which is not permitted. Therefore, the usual constraint for $\hat{a}_i^T p$ should not hold if the ith multiplier has the wrong sign, and a modified set of equality constraints to be satisfied by the search direction should be derived for a reduced active set. If the constraint is believed to be active despite the negative multiplier estimate, $\hat{a}_i^T p$ can be formulated to decrease the ith constraint, for example, according to LC3.

The current strategy is to delete only one constraint with a negative multiplier estimate from the linear equalities at any iteration; the specification of $\hat{a}_i^T p$ for any constrsints that haw negative multiplier estimates even for the modified active set can always be given so that the search direction is a descent direction for the barrier function. Since the current point is not close to x^* in this situation, it seems more satisfying to take the step to a new point, where better multiplier estimates can be obtained, than to continue adjusting the search direction to correspond to information at the current point.

Specification of λ

In a barrier algorithm, where the local influence of the objective function becomes more significant as the barrier parameter is decreased, the vector λ of estimated Lagrange multipliers used in LC4 will always be taken as the least-squares solution cf min $\|g - \hat{A}\lambda\|_2^2$ at the current point. This choice of λ is widely used, in various Lagrangian-type algorithms, because at x* the vector λ* satisfies a least-squares relationship with g(x*) and Â(x*). It is especially appropriate for a barrier function algorithm because the exact relationship of g and Â at any given pcint is significant in the behavior of the algorithm, as will be seen in later discussion of the method's properties.

Calculation of p_N

In calculating the null-space component of the search direction, any tern? included in the Hessian matrix S which is a multiple of $\hat{A}D\hat{A}^T$, for D symmetric and non-singular, is annihilated by application of $\hat{Z}^T$. Therefore, if S is taken as the Hessian of B(x,r), the portion cf S that causes its ill-conditioning for the usual. barrier function method, i.e.,

$$r\hat{A}\begin{bmatrix}\frac{1}{\hat{c}_1^2} & & \\ & \ddots & \\ & & \frac{1}{\hat{c}_m^2}\end{bmatrix}\hat{A}^T,$$

has no effect on the search direction in the barrier trajectory algorithm. The singularity caused by the approach to zero of the active constraint values does not affect the calculation of the search directicn, but only the step length computed during the linear search.

If the current point is close to x*(r) for sufficientiy small r, the choice of either the Hessian of B(x,r) or L(x,λ) for S will give a similar result for p_N’ since the portion of $\nabla^2 B(x,r)$ that remains in computing p_N is given by $G - \sum_{i=1}^{m} \frac{r}{\hat{c}_i}\hat{G}_i$, and $\frac{r}{\hat{c}_i}$ is a first-order approximation to λ_i near $x^*(r)$.

The formulations given for computing p_N do not require that S be positive definite, or even non-singular. However, the specification that p_N minimizes a quadratic function whose Hessian is S in the null space of $\hat{A}$ implies that $\hat{Z}^T S\hat{Z}$ should be positive definite. At x^*, $\hat{Z}^T W\hat{Z}$ is positive definite, where $W = G - \sum_{i=1}^{m} \lambda_i^* \hat{G}_i$. If $\hat{Z}^T S\hat{Z}$ is a good approximation to $\hat{Z}^T W\hat{Z}$, then $\hat{Z}^T S\hat{Z}$ should also be positive definite in a neighborhood of x^*. Nonetheless, as in the penalty trajectory algori-chm, to deal with the inevitable times when $\hat{Z}^T S\hat{Z}$ is not positive definite, the linear system ior p_N is solved with a modified Cholesky algorithm, *so* that the matrix used to compute p_N is always guaranteed to be strictly positive definite (see Gill and Murray, 1972a, for details). If $\hat{Z}^T S\hat{Z}$ is not positive definite, the calculation of the null-space component must be altered in this way; otherwise, the search direction may not be E descent direction for the quadratic function which **it** Is chosen to minimize.

When p_N is written in terms of the matrix $\hat{Z}^T S\hat{Z}$, this latter representation should be interpreted as a matrix that is positive definite and strictly bounded away from singularity, even if the unaltered $\hat{Z}^T S\hat{Z}$ at the current point is singular or indefinite.

Choice of Active Set

A complication inherent in a barrier trajectory algorithm is the decision as tc which constraints are active at x^*, since only their behavior is ultimately significant as the solution is approached. There are numerous possible strategies for choosing the active set of constraints at each iterstion, and details of the particular implementation reported here are given in Chapter 6. The basic principles are that the information generated during the previous iteration and available at the beginning of the current iteration is used to predict the active set. Indications that a particular constraint is not active are: (1) its multiplier estimate i.,negative; or (2) its value is bounded away from zero as the barrier parameter is decreased. The indications that a constraint previously considered inactive is in fact active are: (?·) it is violate6 during the linear search, so that its influence is significant in determining the minimum of the barrier function; or (2) its value is approaching zero as the barrier parameter approaches zero. Using these criteria, an active set of linearly independent constraint gradients is determined at each iteration of the barrier trajectory algorithm. Further details will not be given here, since the method for deciding on the active set is a separable feature of the barrier trajectory algorithm. An incorrect choice of the active set at a particular iteration may slow down local progress, but should not impede ultimate convergence.

Descent Property

Because a linear search to minimize a barrier function is carried out along the trajectory search direction at every iteration, it must be veriiied that the search direction is a descent direction with respect to some barrier function.

If all the current multiplier estimates $\{\lambda_i\}$ are positive, the descent property can be demonstrated for the trajectory search direction, p, where $p = \hat{Q}_1 p_R + \hat{Z} p_N$, such that:

(A) p_R is determined by the linear equalities LC4, i.e.,

$$\hat{A}^T p = \hat{A}^T \hat{Q}_1 p_R = \quad -\hat{c} + \bar{r} \begin{bmatrix} \frac{1}{\lambda_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_m} \end{bmatrix} \quad \text{, where}$$

$\hat{A}^T\hat{Q}_1$ is by definition non-singular, and $\lambda_i > 0$, $i=1,2,\ldots,m$;

(B) p_N satisfies the linear equations $\hat{Z}T S\hat{Z} p_N = -\hat{Z}^T g$, where the matrix $\hat{Z}^T S\hat{Z}$ is strictly positive definite.

The vector λ is the least-squares solution of $\min ||g-\hat{A}\lambda||^2$; the vector g may thus be written as $g = \hat{A} + \hat{Z}v$ for some v, and $\hat{Z}^T g = v$.

As usual, the descent property of p will be verified by showing that the scalar product of p and $\nabla B(x,\hat{r})$ is negative for suitably chosen $\hat{r}$.

For simplicity, we assume without loss of generality that the matrix A is partitioned so that the first m columns contain the gradients of the active constraints, with a similar ordering for the constraint vector. The matrix A can then be written as:

$$A = [\hat{A} \vdots \tilde{A}],$$

where $\hat{A}$ hcs m columns, and $\tilde{A}$ has $(\ell-m)$ columns.

The gradient of the barrier function $B(x,\hat{r})$ is given by:

$$\nabla B(x,\hat{r}) = g - \hat{r}A \begin{bmatrix} \frac{1}{c_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{c_\ell} \end{bmatrix} \quad \text{, or}$$

$$= g - \hat{r}\tilde{A}\begin{bmatrix}\frac{1}{c_1}\\ \\ \\ \\ \frac{1}{c_m}\end{bmatrix} - \hat{r}\tilde{A}\begin{bmatrix}\frac{1}{c_{m+1}}\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \frac{1}{c_\ell}\end{bmatrix} .$$

Substituting the expression $\hat{A}\lambda + \hat{Z}v$ for g, the above becomes:

$$\nabla B(x,\hat{r})' = \hat{A}\begin{bmatrix}\lambda_1 - \frac{\hat{r}}{c_1}\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \lambda_m - \frac{\hat{r}}{c_m}\end{bmatrix} + \hat{Z}v - \hat{r}\tilde{A}\begin{bmatrix}\frac{1}{c_{m+1}}\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \frac{1}{c_\ell}\end{bmatrix} .$$

The scalar product $pT\nabla B(x,\hat{r})$ is then given by:

$$p_R^T\hat{Q}_1{}^T\hat{A}\begin{bmatrix}\lambda_1 - \frac{\hat{r}}{c_1}\\ \cdot\\ \cdot\\ \cdot\\ \cdot\\ \lambda_m - \frac{\hat{r}}{c_m}\end{bmatrix} + p_N{}^T\hat{Z}^T\hat{Z}v + \hat{r}p^T\tilde{A}\begin{bmatrix}\frac{1}{c_{m+1}}\\ \\ \\ \frac{1}{c_\ell}\end{bmatrix} .$$

For $\hat{r}$ sufficiently small and a criterion for choosing the active constraints that guarantees that all inactive constraints are bounded away from zero, the term of the scalar product involving $\tilde{A}$ is of order $\hat{r}$. By construction of $\hat{Z}$, the matrix $\hat{Z}T\hat{Z} = I_{n-m}$, the identity matrix of order n-m. Beceuse p satisfies LC4, so that $a_i{}^T\hat{Q}_1p_R = -c_i + \frac{\bar{r}}{\lambda_i}$, the scalar product can, therefore, be written as follows:

[illegible] (5.5.1),

For. $\bar{r}$ and $\hat{r}$ sufficiently small., the first term is negative because $c_i > 3$, $\lambda_i > 0$; the second term is negative because $\hat{Z}^T S\hat{Z}$ is positivee definite. Consequently, p is a descent direction for $B(x,\hat{r})$.

Throughout the above proof, it waз assumed that $\lambda_i > 0$, $i=1,2,\ldots,m$. The descent property of the trajectory search direction can also be verified when some of the $\{\lambda_i\}$ are negative, if each linear constraint of LC4 corresponding to a negative multiplier is suitsbly altered. The specification of $\hat{a}_i^T p$ for such a constraint can take various forms -- for example, it might be maae "negligible" in order to have no effect on the sign of the scalar product. The key point is that the definition of $\hat{a}_i^T p$ must he altered if $\lambda_i < 0$ because a decrease in $\hat{r}$ only increases the influence of λ_i in the scalar product (5.5.1), given the definition of the multipliers in terms of the vector g. This situation contrasts with the penalty function case, where the influence of a negative multiplier can be made negligible for a sufficiently large value of the penalty parameter.

As mentioned earlier, the search direction that solves the quadratic program QP4 may not be a descent direction for a suitable barrier function at a general point. However, under more restrictive conditions, an appropriate descent property c3n be demonstrated for this choice of trajectory search direction. The required assumptions are: the current point x is close to $x^*(r)$, where r is small enough *so* that $x^*(r)$ is In a near neighborhood of x^*; arid $\lambda_i^* \neq 0$, $i=1,2,\ldots,m$.

To verify the descent property, we must first estimate the magnitudes of the vectors p_R and p_N. The vector p_R is the solution of:

$$\hat{A}^T\hat{Q}_1\ p_R = -\hat{c} + \bar{r} \begin{bmatrix} \frac{1}{\lambda_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_m} \end{bmatrix} \qquad (5.5.2)$$

where $\hat{A}^T\hat{Q}_1$ is by definition non-singular, and $\bar{r} < r$. In a neighborhood of $x^*(r)$, the values $\{\hat{c}_i\}$ are of order r, since $\lambda^*_i = {}^{r}i + O(r)$. The right-hand side of (5.5.2) is thus $O(r)$ for $\bar{r}$ small enough; because $\hat{A}^T\hat{Q}1$ is non-singular, it follows that $||pR|| = O(r)$. Since the columns of $\hat{Q}_1$ are orthogonal, $||\hat{Q}_1p_R||$ is also of order r. The vector pN is given by:

$$p_N = -(\hat{Z}^TS\hat{Z})-1\ (\hat{Z}^Tg + \hat{Z}^TS\hat{Q}_1p_R). \qquad (5.5.3)$$

The matrix $\hat{Z}^TS\hat{Z}$ is guaranteed to be strictly positive definite, and hence $||(\hat{Z}^TS\hat{Z})^{-1}||$ is bounded. The norm of the vector $\hat{Z}^TS\hat{Q}_1p_R$ satisfies:

$$||\hat{Z}^TS\hat{Q}_1p_R|| \leq ||\hat{Z}^TS||\ ||\hat{Q}_1p_R||\ .$$

The norm of $\hat{Z}^TS$ is bounded in a neighborhood of $x^*(r)$ for any of the given choices for S. Since $||\hat{Q}1pR||$ has 'seen shown to be $O(r)$, it follows that $||\hat{Z}^TS\hat{Q}_1p_R|| = O(r)$.

To estimate $||\hat{Z}^Tg||$, consider the function $e(x)$ representing the least-squares error,

$e(x) = \min ||g-\hat{A}\lambda||^2 = ||\hat{Z}^Tg||^2$, for g, $\hat{A},\hat{Z}$ in a neighborhood of $x^*(r)$. Because the matrix $\hat{A}$ has full, constant rank in a neighborhood of $x^*(r)$, for r small enough, arid the second derivatives of F and $\{c_i\}$ are continuous by assumption, the function $e(x)$ can be written:

$$e(x) = ||g-\hat{A}(\hat{A}^T\hat{A})^{-1}\hat{A}^Tg||^2,$$

some neighborhood of $x^*(r)$ (see Golub and Pereyra, 1973).

Since at $x^*(r)$, $||\hat{Z}^T g||_2^2 = e(x^*(r)) = O(r^2)$, the relationship $||\hat{Z}^T(x^*(r)+\delta)\, g(x^*(r)+\delta)||_2^2 = e(x^*(r)+\delta) = O(r^2)$ will hold for sufficiently small $||\delta||$, by continuity of $e(x)$. Therefore $||\hat{Z}^T g||$ is $O(r)$ in an appropriate neighborhood of $x^*(r)$. From (5.5.3), $O(||p_N||) = O(||(\hat{Z}^T S\hat{Z})^{-1}||)\, O \max(||\hat{Z}^T S\hat{Q}_1 p_R||, ||\hat{Z}^T g||)$, and thus we see that $||p_N||$ is $O(r)$. Since $||p||^2 = ||p_N||^2 + ||p_R||^2$, $||p||$ must also be $O(r)$.

Given these estimates of $||p_R||$ and $||p_N||$, we consider the scalar product of the search direction with $\nabla B(x,\hat{r})$, given by:

$$p^T \quad (g - \hat{A}\begin{bmatrix} \frac{\hat{r}}{c_1} \\ \vdots \\ \frac{\hat{r}}{c_m} \end{bmatrix} - \tilde{A}\begin{bmatrix} \frac{\hat{r}}{c_{m+1}} \\ \\ \frac{r}{c_\ell} \end{bmatrix}) .$$

Replacing g by its representation $\hat{A}\lambda + \hat{Z}v$, and p by $\hat{Q}_1 p_R + \hat{Z}p_N$, we obtain as the scalar product:

$$p_R^T \hat{Q}_1^T \hat{A} \begin{bmatrix} \lambda_1 - \frac{\hat{r}}{c_1} \\ \vdots \\ \lambda_m - \frac{\hat{r}}{c_m} \end{bmatrix} + p_N^T v - p^T \tilde{A} \begin{bmatrix} \frac{\hat{r}}{c_{m+1}} \\ \\ \frac{\hat{r}}{c_\ell} \end{bmatrix} .$$

Substituting the value of each component of $\hat{A}^T \hat{Q}_1 p_R$, the expression becomes:

$$\sum_{i=1}^{m} (-c_i + \frac{\bar{r}}{\lambda_i})(\lambda_i - \frac{\hat{r}}{c_i}) + p_N^T v - p^T \tilde{A} \begin{bmatrix} \frac{\hat{r}}{c_{m+1}} \\ \\ \frac{\hat{r}}{c_\ell} \end{bmatrix} .$$

Each term $(-c_i + \frac{\bar{r}}{\lambda_i})(\lambda_i - \frac{\hat{r}}{c_i})$ is $O(r)$ and negative, for $\bar{r}$, $\hat{r}$ sufficiently small, since each λ_i is positive close to $x^*(r)$. The term $p_N^T v$ is $O(r^2)$, since $||p_N||$ is $O(r)$, and $v = \hat{Z}^T g = O(r)$; the term involving $\tilde{A}$ is $O(\hat{r})$, and $\hat{r} < r$. Therefore, the magnitude and negative sign of the first term dominate the scalar product, and the trajectory search direction that solves QP4 is a descent direction for $B(x,\hat{r})$ under the stated assumptions. It should be emphasized, however, that this descent property does not hold at a general point; even if all the estimated multipliers are positive.

Local Convergence Properties

A unit step can be taken along the trajectory search direction if it will maintain feasibility. For any non-zero $\bar{r}$, the trajectory search direction satisfies:

$$\hat{A}^T p = -\hat{c} + r \begin{bmatrix} \frac{1}{\lambda_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{\lambda_m} \end{bmatrix} .$$

The expansion of $\hat{c}_i$ about the current point can be written:

$$\hat{c}_i(x + p) = \hat{c}_i + \hat{a}_i^T p + \frac{1}{2} p^T \hat{G}_i p + O(||p||^3) . \qquad (5.5.4)$$

Substituting the value $-\hat{c}_i + \frac{r}{\lambda_i}$ for $\hat{a}_i^T p$ in (5.5.4), we obtain:

$$\hat{c}_i(x + p) = \frac{\bar{r}}{\lambda_i} + \frac{1}{2} p^T \hat{G}_i p + O(||p||^3) .$$

For x sufficiently close to $x^*(r)$, the step taken by the trajectory algorithm is $O(r)$ in magnitude, and thus the term $(\frac{1}{2} p^T G_i p)$ will be $O(r^2)$. The term $\frac{\bar{r}}{\lambda_i}$ is $O(\bar{r})$, and, consequently, if $O(\bar{r}) > O(r^2)$, $\hat{c}_i(x + p)$ will be positive, and a unit step along p will maintain feasibility.

The trajectory search direction that solves the quadratic programing formulation QP4 for sufficiently small $\bar{r}$ can be made arbitrarily close to the step taken 'cy an analogous linearly constrained Lagrangian algorithm if the same matrix S is used in both methods. This result is demonstrated by the same reasoning given in Section 5.4.3 for the penalty trajectory algorithm; the right-hand side of the linear constraints of QP4 is an O ($) perturbation of the constraints in a Lagrangian method, and thus the search direction generated by the barrier trajectory algorithm differs from the Lagrangian step by $O(\bar{r})$. Therefore, close to x*, with a judicious choice of $\bar{r}$ that maintains feasibility, the barrier trajectory algorithm based on QP4 may achieve the local convergence rate of a linearly constrained Lagrangian algorithm if $\bar{r}$ is small enough so that the $O(\bar{r})$ perturbation is insignificant with respect to the terms that determine the rate of convergence. Further comments on the local convergence of the barrier trajectory algorithm are giver, in Section 5.6.

When the vector p_N in the trajectory search direction is computed to satisfy the alternative formulation E3, i.e.,

$$2^{T}S\hat{Z}p_N = - \hat{Z}^T g, \qquad (5.5.5)$$

and S is given by the Hessian of an approximate Lagrangisn function, the null-space component of the trajectory search direction, $\hat{Z}p_N$, becomes an arbitral-ily close approximation to the null-space component of the Newton step to the minimum of $B(x,\hat{r})$ for % sufficiently small. As in the penalty case, the results of Murray (1971) give an explicit and highly accurate approximation of the inverse of the Hessian matrix $\nabla^2 B(x^*(\hat{r}),\hat{r})$:

$$(\nabla^2 B(x^*(\tilde{r}),\tilde{r})^{-1} \doteq \tilde{r}\hat{A}(\hat{A}^T\hat{A})^{-1}\begin{bmatrix} \frac{1}{\lambda_1^2} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \frac{1}{\lambda_m^2} \end{bmatrix}(\hat{A}^T\hat{A})^{-1}\hat{A}^T + Z(\hat{Z}^T(G-\sum_{i=1}^{m}\lambda_i\hat{G}_i)\hat{X})^{-1}\hat{Z}^T,$$

where $\{\lambda_1\}$ are the estimated Lagrange multipliers.

An approximation to the Newton step, $\tilde{p}$, to the minimum of $B(x,\tilde{r})$ can be written as:

$$\tilde{p} \doteq -(\nabla^2 B(x^*(\tilde{r}),\tilde{r}))^{-1} \left(g-\tilde{r}\hat{A}\begin{bmatrix} \frac{1}{c_1} \\ \\ \frac{1}{c_m} \end{bmatrix}\right),$$

where the contribution of the inactive constraints has been ignored. Writing g in term of its expansion as $\quad + \hat{Z}v$, $\tilde{p}$ separates into twc orthogonal components, and the null-space component of p is approximately given by:

$$\hat{Z}\tilde{p}_N \doteq -\hat{Z}(\hat{Z}^T(G-\sum_{i=1}^{m}\lambda_i\hat{G}_i)\hat{Z})^{-1}v,$$

which is exactly equal to the null-space component of the trajectcry direction given by (5.5.5) if $S = G - \sum_{i=1}^{m}\lambda_i\hat{G}_i$.

These results show that it is possible to demonstrate local convergence' for the 'carrier trajectcry algorithm. However, as noted for the penalty trajectory algorithm, local convergence does not necessarily imply a robust or generally effective algorithm. The strength cf the 'carrier trajectory algorithm is based on its derivation through the logarithmic barrier function, in terms of properties that do not critically depend on being in a neighborhood of x^*.

5.6 Summary

The penalty and barrier trajectory algorithms that have been presented are 'cased on the idea of using the properties of the quadratic penalty function and the logarithmic barrier function to characterize the search directior et each iteration. The measure of progress throughout is provided by a suitable decrease in either a penalty or barrier function, with a forcing parameter that may be altered at every step.

If the trajectory search direction solves the appropriate quadratic programming problem, the local rate of convergence of the trajectory algorithms is not limited to the rate of convergence of the multiplier estimates, given a suitable choice of the penalty or barrier parameter. The analysis of local convergence for the trajectory algorithms follows that given in Section 4.8 for the linearly constrained Lagrangian algorithms, and the discussion here will ccnsequently be abbreviated. The method to be considered is the penalty trajectory algorithm for equalities, where the exact Hessians of F and $\{c_i\}$, i=1,2,...,m, are available at the current point.

Let $\bar{x}$ and $\bar{\lambda}$ be the current estimates of x^* and A^*. The step $\bar{p}$ to the next estimate, $\bar{\bar{x}}$, is the solution of the quadratic programming problem:

$$\text{minimize } \frac{1}{2} p^T(\bar{G} - \sum_{i=1}^{m} \bar{\lambda}_i \bar{G}_i)\, p + p^T\bar{g}$$

$$\text{subject to } \bar{A}^T p = -\bar{c} - \frac{\bar{\lambda}}{\rho} \quad ,$$

where "$\bar{\alpha}$" denotes the function "α" evaluated at $\bar{x}$. The new multiplier estimate $\bar{\bar{\lambda}}$ will be given by the multipliers of the quadratic programming problem.

Define $\bar{e} = \bar{x} - x^*$, $\bar{\bar{e}} = \bar{\bar{x}} - x^*$, $\bar{A} = \bar{\lambda} - \lambda^*$, and $\bar{\bar{\Delta}} = \bar{\bar{\lambda}} - \lambda^*$. Since $\bar{\mathbf{p}}$ satisfies $\bar{A}^T\bar{\mathbf{p}} = -\bar{c} - \frac{\bar{\lambda}}{\rho}$, by writing $\bar{p}$ as $\bar{\bar{e}} - \bar{e}$, we obtain:

$$\bar{A}^{T}\bar{\bar{e}} = -\bar{c} + \bar{A}^{T}\bar{e} - \frac{\bar{\lambda}}{\rho} \ . \qquad (5.6.1)$$

The Taylor series expansion of c about x* gives:

$$\bar{c} - \bar{A}^T\bar{e} + O(\|\bar{e}\|^2) = c(x^*) = 0. \qquad (5.6.2)$$

Substituting (5.6.2) in (5.6.1), the result is:

$$\bar{A}^T\bar{\bar{e}} = O\max\left(\frac{1}{\rho}, \|\bar{e}\|^2\right). \qquad (5.6.3)$$

The estimate $\bar{\bar{\lambda}}$ is defined by:

$$\bar{A}\bar{\bar{\lambda}} = \bar{g} + \left(\bar{G} - \sum_{i=1}^{m} \bar{\lambda}_i\bar{G}_i\right)\bar{p} \cdot \qquad (5.6.4)$$

Using the same procedure as In Section 4.8, we expand g and $\{a_i\}$, $i=1,2,\ldots,m$, about x*, substitute the expansions in (5.6.4), and regroup, yielding:

$$\bar{A}\bar{\bar{\Delta}} = \bar{W}\bar{\bar{e}} + O(\|\bar{\Delta}\|\ \|\bar{e}\|) + O(\|\bar{e}\|^2), \qquad (5.6.5)$$

where $\bar{W} = G - \sum_{i=1}^{m} \bar{\lambda}_i\bar{G}_i$.

The derived relationships (5.6.3) and *(5.6.5)* for $\bar{\bar{e}}$ and $\bar{\bar{\Delta}}$ may then be written as:

$$\begin{bmatrix} \bar{W} & -\bar{A} \\ \bar{A}^T & 0 \end{bmatrix} \begin{bmatrix} \bar{\bar{e}} \\ \bar{\bar{\Delta}} \end{bmatrix} = \begin{bmatrix} O(\|\bar{\Delta}\|\ \|\bar{e}\|) + O(\|\bar{e}\|^2) \\ O\max\left(\frac{1}{\rho}, \|e\|^2\right) \end{bmatrix}. \qquad (5.6.6)$$

For $\bar{A}$ of full rank, the result (5.6.6) indicates that if the penalty Parameter ρ is adjusted near x* so that $O(\frac{1}{\rho}) = O(\|e\|^2)$, the trajectory algorithm will maintain the local rate of convergence of the analogous linearly constrained Lagrangian algorithm. A similar result applies io the penalty algorithm for inequalities , For the barrier trajectory algorithm, a result analogous to (5.6.6) holds if the barrier parameter, $\bar{r}$, which defines the right-hand side of the linear constraints LC4 satisfied by the search direction, is chosen so that $\mathcal{O}(\bar{r}) = \mathcal{O}(||\bar{e}||^2)$. However, as noted in Section 5.5, the maintenance of feasibility must also be considered in adjusting the 'carrier parameter in the barrier trajectory method. If $\bar{r}$ is too Small, the step given by the linear constraints may yield a non-feasible point, and thus a limit to the rate of local convergence may be imposed by the requirement of feasibiliiy.

For the inequality constrained problem, the trajectory algorithms generate information that is useful in identifying the correct active set. With the penalty algorithm, the Lagrange multipliers of the quadratic programming sub-problem may allow a tetter prediction of the active set. Moreover, because cf the underlying reliance on the properties of a penalty function, the convention of choosing the currently violated set of constraints as a prediction of the active set remains reasonable.

For the barrier algorithm, the active set is predicted throughout by monitoring the signs of the multiplier estimates and the decrease in constraint values, and by noting which constraints are violated during eacn iinear search. Near the solution, when the quadratic programing formulation is used to compute the search direction, the multipliers of the sub-problem provide a higher-order estimate of λ^*, and hence of the active set.

The behavior of the penalty and carrier trajectory algorithms predicted by the theoretical analysis of this chapter has been verified experimentally by the numerical results presented in Chapter 6, which demonstrate that the trajectory algorithms are quite successful in solving a variety of problems.

CHAPTER 6. Implementation of Algorithms and Numerical Results

6.1 Discussion of Comparison of Optimization Algorithms

6.2 Choice and Implementation of Algorithms

6.3 Discussion cf Test Problems

6.3.1 Non-feasible Methods

6.3.2 Feasible Methods

CHAPTER 6

IMPLEMENTATION OF ALGORITHMS AND NUMERICAL RESULTS

6.1 ~~Discussion~~ of Comparison of Optimization Algorithms

The complications inherent in making an adequate comparison amocg different algorithms for certain classes of problems are well known (see Lyness and Kaganove, 1976; Gill an3 Murray, 1974b) The difficulties in comparicg optimization methods include the following:

(1) the quantities used to measure performance may vary widely;

(2) the details of implementation may have an enormous effect on the results, but are usually not published;

(3) most published tests are limited to a small number of starting points (often only one), and the values chosen for critical parameters cf some aspects of the methods may not be specified . Furthermore, the examples given do not usually proviae sufficient variety to test all relevant aspects of the methods considered.

The factcrs (2) and (3) may be overcome to a large extent 'cy publication cf all significant details of the particular implementations used in the comparison, by increasing the number of test problems and starting points, and by systematic experimentation to determine the effects of variation in the significant parameters of each method.

The difficulty posed by (1) is more complicated, because there is simply no universal measure of performance for an optimization algorithm. Assuming that the methods do not fail, the relative merit of different algorithms is entirely dependent on the characteristics of the problem to be solved. Therefore, the question f which algorithm is "best" car',-

not be answered without some quite specific assumptions about the problem in question.

For example, it is sometimes stated that. the measure of comparison should be the number of "equivalent function evaluations" required for convergence, where a gradient is considered to be worth n function evaluations, and a Hessian is counted as $\frac{n(n+1)}{2}$ function evaluations. This weighting is obviously derived from the notion of using finite differences to approximate the gradient and Hessian, and it might be a reasonable measure if these procedures were always followed. However, it is nonsensical to impose this standard to compare algorithm on a problem where the gradient or Hessian can be evaluated much more rapidly than one would expect from the finite difference analogy. For such a problem, the number of "equivalent function evaluations" is highly misleading.

A similar criticism can 're made when the comparison is based on the elapsed computer time. There are obvious defects in using this measurement to compare different implementations of algorithms, because cf the very large variation in execution time that depends on festures of the particular computer, the compiler, and program structure, rather than on the methcd itself. A more fundamental inadequacy is that this measure does not distinguish between the time spent carrying out the fixed procedures of each method, which normally depend only on the dimensionality of the problem, and the time required to evaluate the problem functions. If the problem functions can be evaluated rapidly, a naive method with a small amount of housekeeping per iteration may appear "better" than a more powerful and robust method, since the former might use less computer time, despite requiring many more iterations and function evaluations to achieve the desired accuracy. However, the comparison would be reversed if the time spent calculating the problem functions completely dominates the program overhead, since in this case a saving of computer time will result from using the more sophisticated method that converges with fewer function evaluations.

A further factor worth emphasizing is that the inclusion of safeguards to ensure reliability and numerical stability always makes programs more lengthy and complicated, so that the "simplicity" of a prcgram may merely imply that it is non-robust. The problems in a comparison should include examples that pose as many difficulties as possible, in order to determine the weaknesses of the methods tested. Otherwise, a program where care has been taken to protect against failure and instability may seem to be less desirable than a naive unsafeguarded program that might fail completely on some examples.

Given these considerations, the comparison to be given here is not claimed to be comprehensive. Rrther, the emphasis hzs been to examine in detail the behavior of each method, and to highlight the significant computational differences in the theoretical algorithm.

For each algorithm, the work per iteration will be discussed, in terms of a description of the numerical procedures followed, including relevant functions of dimensionality. The following information will be given for each test example:

(1) number of iterations;

(2) number of evaluations of F and $\{c_i\}$;

(3) number of evaluations of g and $\{a_i\}$;

(4) number of evaluations of G and $\{G_i\}$.

In any given comparison, all algorithm use the same termination criteria, and the free parameters associated with the analogous steps of eacn method are th̶ same (for example: the criteria for terminating tne linear search).

5.2 Choice and Implementation of Algorithms

Choice of Methods

The algorithms implemented were selected to form a reasonably general set of approaches to solving nonlinearly constrained optimization problems; however, certain classes of algorithms were excluded *a priori*. For example, Linearly constrained Lagrangian algorithms can easily be made to fail if implemented in precisely the forms given in Section 4.8; these algorithms can be varied to improve robustness -- e.g., by adding a feature to enforce decreasing infeasibility, by re-formulating an unbounded sub-problem -- but such modifications are not the primary concern in this dissertation. A continuous augmented Lagrangian algorithm is not included because of the drawback that second derivatives of the problem functions are required to use a first-order technique to solve the unconstrained sub-problem (see Section 4.5).

A class of algorithms not considered here is that of "generalized reduced gradient" acd "projected gradient" methods (see Rcsen, 1961; Abadie and Carpentier, 1969; Sargent and Murtagh, 1973; Lssdon et al., 1974). Especially in their latest forms, these methods have strong similarities to algorithms that solve a quadratic program to obtain the search direction. The main distinguishing feature of such algorithms is the incorporation of special steps to retain feasibility; these steps have the effcct of generating a sequence of iterates that lie close to the boundary cf the feasible region.

The following methods were implemented:

(1) For the general nonlinearly constrained problem

(a) quadratic penalty function method (Section 2.2);

(b) sequential augmented Lagrangian method (Section 4.3);

(c) progressive augmented Lagrsngian method (Section 4.4);

(d) penalty trajectory method (Section 5.4);

(2) For problems where strict feasibility is to be maintained

(a) logarithmic barrier function method (Section 2.4);

(b) barrier trajectory method (Section 5.5).

Algorithms with Unconstrained Sub-problem

A single basic program structure can be used to implement all the algorithms involving an uncoastrained sub-problem at each iteration, since the search direction may be computed using one of the methods for unconstrained cptimization described in Section 1.4. Two unconstrained algorithms have beer, used in the programs -- a Newton-type method, and a quasi-Newton method. With either of these methods, the search direction is generally given by the solution of a system of linear equations of the form:

$$Bp = - b, \tag{6.2.1}$$

where b is the gradient of the function to be minimized. The Newton-type algorithm implemented (see Gill and Murray, 1972a) overcomes the deficiencies of Newton's method (*see* Section 1.4) by always using a positive definite matrix B in (6.2.1). The matrix B is obtained by attempting to form the Cholesky factorization of the exact or approximated Hessian of the given function. As the factorization is computed, **it** is possible to determine whether the original Hessian is strictly positive definite. If *so*, the unaltered matrix serves as the matrix B in (6.2.1). However, **if** the Hessian is indefinite or not sufficiently positive definite, the matrix actually factorized is the sum of the original Hessian plus a pcsitive diagonal matrix, whose elements are chosen so that the modified matrix is guaranteed to be positive definite.

The implemented quasi-Newton algorithm is a "revised quasi-Newton method" (see Sill and Murray, 1972b), in which the Cholesky factors of an approximation to the Hessian are updated at each iteration, and the resulting positive definite matrix serves as B in (6.2.1).

In a barrier function algorithm, the search direction at each iteration is chosen by solving the usual set of linear equations. However, the need to maintain strict feasibility leads to additional logic in the program structure -- for esample, the initial step taken during the linear search is estimated by a special method (see Section 3.3.1).

Form of Unconstrained Function

For simplicity, the notation to be used during this discussion will be that of the equality constrained problem, P1. The observations to be given will apply to the sane methods for the inequality constrained problem, P2, 'cut with an explicit determination cf the "active set" of constraints at each point. Therefore, multipliers correspond only to "active" constraints, and only "active" constraints ere included in any penalty term.

For a penalty function method, a sequential augmented Lagrangian method, and a progressive Lagrangian method, the unconstrained function to be minimized at every iteration has the form of an augmented Lagrangian function:

$$F(x) - \lambda^T c(x) + \frac{\rho}{2} c(x)^T c(x).$$

With a penalty function method, the vector λ is always taken as zero. The value of ρ is fixed throughout each unconstrained minimization, and successively increased until suitable convergence criteria are satisfied. In a sequential augmented Lagrangian method, the vector λ is the current estimate cf the Lagrange multipliers, and is fixed daring each complete unconstrained minimization. The parameter ρ is fixed during each minimization, and may be increased if there is insufficient decrease in infeasibility.

With a progressive method, the vector λ is updated or re-computed at the beginning of each iteration (or with scme other frequency). The parameter ρ in a progressive algorithm is increased if neither $||c||$ nor $||g-A\lambda||$ decreased sufficiently during the preceding iteration; however, some other criterion for modifying ρ can also be used. **It** should be emphasized that in a progressive algorithm, each step is directed toward minimizing a different function, since in general λ will vary from iteration to iteration. Therefore: the use of a quasi-Newton method to approximate the Hessian may be questionable if the multiplier estiaates and/or penalty parameter are changing rapidly.

Calculation cf Multiplier Estimates

The first-order estimate of the Lagrange multpliers at a point is given by the least-squares solution of min $||g-A\lambda|$ This problem is solved through the implicit calculation of the orthogcnal factorization of A (see Peters and Wilkinson, 1970, for complete details). A sequence of Householder transformations is constructed, to be, applied on the left to reduce A to upper-triangular icrm. Each transformation may be stored as a vector.

If the matrix A is not of full. rank, many of the frequently proposed formulas for computing the multipliers will fail --.for example, the expression $(A^TA)^{-1}A^Tg$ is undefined if A is rank-deficient. When solved by use of orthogonal transformations, a least-squares solution can still be defined, although it is not unique. However, Its interpretation as a set of multipliers is not completely straightforward.

Higher-order multiplier estimates are computed as the least-squares solution of

$$AX = S\tilde{p} + g, \qquad (6.2.2)$$

where S is an approximation to the Hessian of the Lagrangian function, and $\tilde{p}$ is the solution of the quadratic programming problem:

$$\min \frac{1}{2} p^TSp + p^Tg \qquad (6.2.3)$$

$$\text{subject to } A^Tp = -c.$$

The solution $\tilde{p}$ is computed 3s the sum of two orthogonal components, as described in the next section for the trajectory search direction. The motivation for this definition of the multipliers is given in Section 4.3.

Program Structure for Trajectory Algorithms

The search direction, p, for the trajectory algorithms is computed as the sum of two orthogonal components, as described in Chapter 5. To simplify the notation, the discussion will concern primarily the penalty trajectory algorithm for equality constrained problems.

We shall consider in detail only the case where the matrix A is cf full rank. The estimate of the multipliers, λ, required to define the right-hand side of the linear equality constraints, is given by the usual least-squares solution of min $||g-A\lambda||^2$ at the current point.

To compute the matrices Q_1 and Z, the explicit product is formed of the sequence of orthogonal transformations applied to A to solve the least-squares problem. The resulting n by n orthogonal matrix Q is such that

$$QA = \left[\begin{array}{c} R \\ \cdots \\ 0 \end{array}\right] \begin{array}{l} \}\; m \\ \\ \}\; n-m \end{array} ,$$

or, equivalently,

$$A^T = \left[\begin{array}{c:c} L & 0 \end{array}\right] Q \text{ (the "LQ" factorization of } A^T\text{).}$$

The matrices Q_1 and Z can be obtained from partitioning Q. Let Q be partitioned such that

$$Q = \left[\begin{array}{c} Q_1^T \\ \hdashline Z^T \end{array}\right] \begin{array}{l} \}\; m \\ \}\; n-m \end{array} ,$$

then

$$A^T \left[\begin{array}{c:c} Q_1 & Z \end{array}\right] = \left[\begin{array}{c:c} L & 0 \end{array}\right] ,$$

and Q_1 and Z satisfy the required properties.

The component of the search direction in the range of A, $Q_1 p_R$, is uniquely determined by the linear equalities of the trajectory algorithm:

$$A^T p = -c - \frac{\lambda}{\rho} ,$$

since, by construction of Q_1,

$$A^T Q_1 p_R = L p_R = -c - \frac{\lambda}{\rho} \qquad (6.2.4)$$

and L is non-singular. The vector p_R is thus the unique solution of the m by m linear system (6.2.4).

The vector p_N is the solution of a linear system of the form:

$$Z^T S Z p_N = -Z^T w , \qquad (6.2.5)$$

for some matrix S and vector w. The matrix S is normally an approximation to the Hessian of the penalty or Lagrangian function. If second derivatives are available, the matrix $Z^T SZ$ can be formed explicitly. If only first derivatives are available, the matrix $Z^T SZ$ can be approximated directly by differencing the appropriate gradients along the (n-m) columns of Z; this procedure requires only (n-m) gradient evaluations, rather than the n gradient evaluations generally needed to form a finite-difference approximation to the full Hessian. The linear system (6.2.5) is solved by a modified Cholesky algorithm (see Gill and Murray, 1972a); if the original matrix is not sufficiently positive definite, the algorithm forms the Cholesky factors of a perturbed matrix that is guarantees to be strictly positive definite.

For the trajectory algorithms for inequalities, the above procedure is also followed to compute the search directicn, except that only the "active" constraints are included in the definition of the linear equality constraints.

When the matrix A is not of full rank, the matrices Q_1 and Z may be obtained from the transformations applied to A, but their dimensions will be altered.

Linear Search

The linear searches in all the non-feasible methods are the implementations by Gill and Murray (1974a) of safeguarded step length algorithms, based on either cubic or parabolic interpolstion with safeguards to assure termination and stability. The feasible methods use special line searches for the logarithmic barrier function, which are the step length algorithms mentioned above, including the option of fitting appropriate special functions rather than polynomials (see Chapter 3).

Active Set in Barrier Trajectory Algorithm

The current procedure for choosing the initial "active" set in the barrier trajectory algorithm is the following, where m represents the number of "active" constraints, and I denotes the set of indices of the "active" constraints.

```
(m ← 0; I ← φ (null set); $\alpha_u$ ← usuai initial step to
 be taken along trajectory search direction;)
while m < n
   repeat ⟨
(1) compute the trajectory search direction, p, with the set
    I of active ccnsrraints;
(2) for all i ∈ I, compute $\hat{i}$, the index for which $\hat{\alpha} = \alpha_{\hat{i}} =$
    $\min_{\substack{i \in I \\ a_i^T p < 0}} \left( \frac{-c_i}{a_i^T p} \right)$
    $a_i$ is linearly independent of $\{a_j\}$, $j \in I$.

    If $a_i^T p > 0$ for all $i \in I$, exit;
```

NOTE: This procedure yields $\hat{\alpha}$ as the first-order estimate of the smallest positive step along p to a zero of a currently "inactive" constraint.

```
(3) If $\hat{\alpha} < \alpha_u$ $(I \leftarrow I \cup \{\hat{i}\};\ m \leftarrow m + 1;\rangle$
$\rangle$
```

The "active set" may be updated at every iteration, based on the following general criteria . If any "inactive" constraints are violated during the linear search, the one violated nearest to the starting point is added to the "active set". A check is also made to determine whether any "inactive" constraint is becoming "small", or whether any allegedly "active" constraint is bounded away from zero. These tests are obviously highly dependent on the current value of the barrier parameter, and cn an implicit scaling (i.e ., the definition of "small").

At the beginning of each iteration, the multiplier estimates are computed for the "active set" If any of the estimated multipliers are negative, the constraint corresponding to the most negative multiplier is deleted from the sctive set- As mentioned in Section 5.5, the usual definition of the trajectory search direction should not apply if the predicted multiplier is negative. Only one such constraint is deleted per iteration, following whicn the multiplier estimates are updated for the reduced active set. The specification of $a_i^T p$ for any constraint with a negative multiplier in the modified set is set to $(-\epsilon)$ for negligible E.

The ultimate decision about the active set must be determined by the constraint values, rather than the signs of the multiplier estimates. Along the trajectory of a barrier function method, the gradient of F is always a non-negative linear combinatio, of all the constraint gradients, so that the distinction between "active" and "inactive" cannot be based on the signs of the estimated multipliers.

General Notes on Programs

The programs are written using MORTRAN, a FORTRAN language preprocessor developed at the Stanford Linear Accelerator Center (see Cook and Shustek, 1975); the code actually executed is written in ANSI FORTRAN. All runs were made on the IBM 360/91 or 370/168 (the Triplex System at SLAC), with double precision arithmetic throughout, i.e., machine precision is 16^{-13}, or 2.220 x 10^{-16}, with a range for floating point numbers of (5.4 x 10-79, 7.2 · 10^{75}).

6.3 Discussion cf Test Problems

The set of meaningful test problems in the literature is very small; although some previously published examples are included, it was felt necessary to generate some new examples as well. The results presented are typical of those obtained on a wider group of problems, with varied starting points and parameters.

Examples with a large number of linear Constraints were deliberately not chosen, because many algorithms are available to take explicit advantage of known linearity in the constraint functions (see Gill and Murray, 1972a; Gill and Murray, 1974b). Furthermore, linear constraints do not test the difficulties that arise from varying constraint curvature and gradients, and many aspects of the computation simplify greatly in the linear case (for example, the predicted first-order step to a ccnstraint zero is exact).

All the results tc be reported are based on implementation; that use the exact second derivatives of the problem functions. Other programs have been developed that require only first derivatives (or can approximate second derivatives by finite differences), but many questions remain unanswered for some of these methods. For example, there is some uncertainty as to the meaning of a quasi-Newton approximation to the Hessian of a function that changes at every iteration when the multipliers are re-computed. In any case, the results obtained thus far for the gradient cases are consistent with the trends indicated here.

Each example will be discussed briefly, and any notable features will be mentioned. In all cases, the number of iterations is equal to the number of Hessian evaluations, so that the latter will not be listed separately. The parameter η is the criterion for terminating the linear search, and is defined in Section 3.3.

6.3.1 Non-feasible Methods

The sequential augmented methods use the zero vector as the initial approximation to the Lagrange multipliers, so that the first minimization of the sequential method Is identical to that of the penalty function method.

The first-order multiplier update for the sequential methods is given by the Hestenes/Powell correction described in Section 4.3; the first-order multiplier estimate for a progressive method is the least-squares solution of min $||g-A\lambda||^2$. The second-order multiplier update and estimate are described in Section 6.2.

The linear search parameter η is 0.5 for all the non-feasible methods. Each unconstrained sub-problem terminates when the gradient of the unconstrained function satisfies $||\text{grad}|| \leq 10^{-7}$; final convergence is attained when $||c|| \leq 10^{-7}$ for the active constraints.

EXAMPLE 1:

$$\min x_1 x_2^2$$

subject to

$$c_1 = x_1^2 - x_2^2 + 2 \geq 0.$$

$$x^* = (-.816497$$

$$-1.15470)$$

$$\lambda^* = (.816497)$$

This two-dimensional example is included so that the convergence paths for different methods can be plotted (Figure 6.1). Of particular interest is the fact that the point located by the sequential algorithm after the initial minimization is not on the "penalty side" of the constraint.

Example 1

Start (.5, .5)

	Iterations	Functions ani Gradients
Penalty Method:		
ρ = 1.	6	8
10	4	4
10^2	3	3
10^3	2	2
10^4	2	2
10^5	1	1
TOTAL	10	

	Iterations	Functions and Gradients
Sequential Method, First-order multiplier update:		
p = 1.0	6	8
	5	5
	4	4
ρ = 10	3	3
	2	2
	1	1
TOTAL	22	24
Progressive Method, First-order multiplier estimate:		
ρ = 1	9	9
Trajectory Method:	6	7

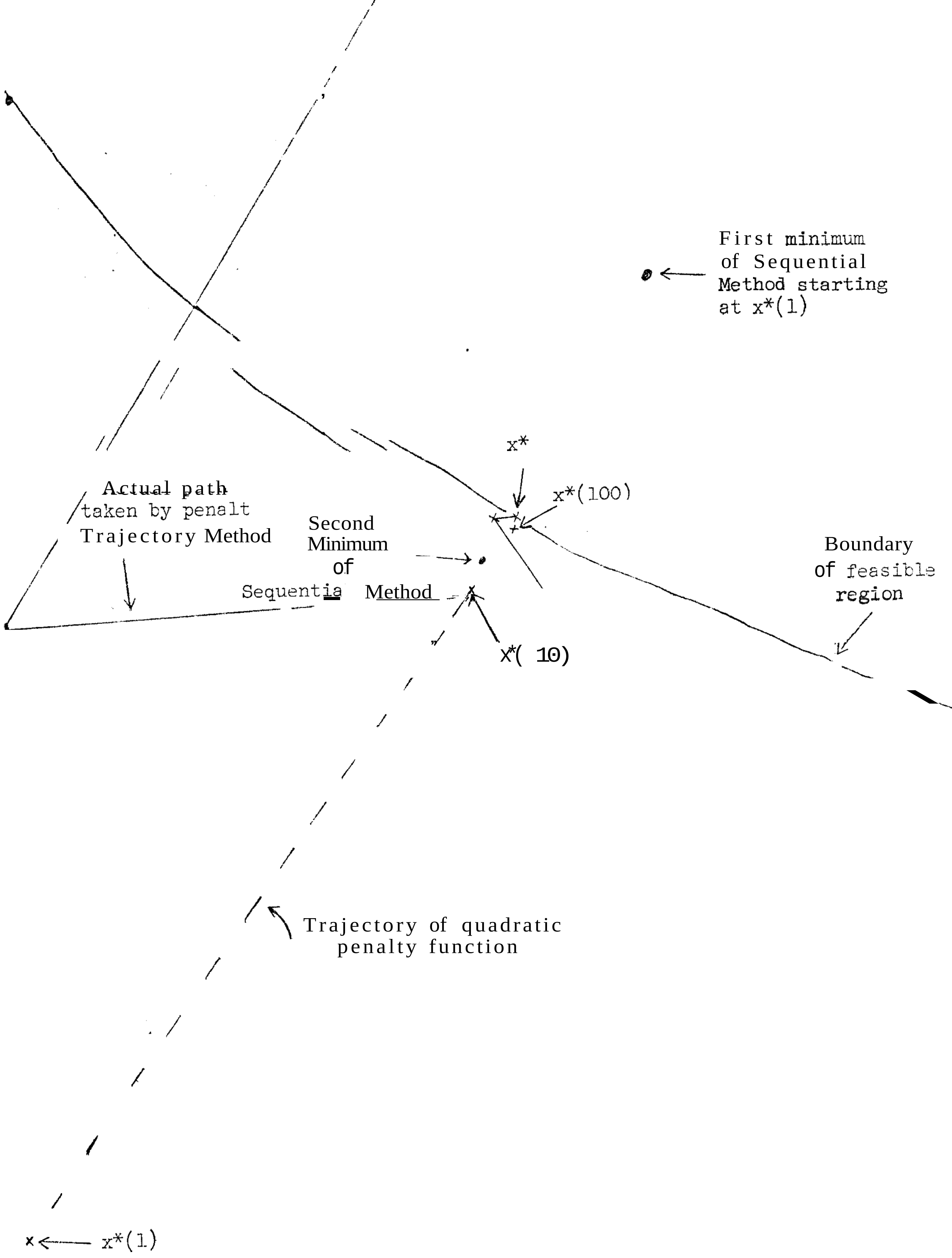

First minimum
of Sequential
Method starting
at x*(1)
x*
x*(100)
Actual path
taken by penalt
Trajectory Method
Second
Minimum
of
Sequentia Method
Boundary
of feasible
region
x*(10)
Trajectory of quadratic
penalty function
x*(1)

FIGURE 6.1

EXAMPLE 2:

$$\min \quad 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

subject to

$$c_1 = x_1 + x_2 + x_3$$

$$c_2 = 8x_1 + 14x_2 + 7x_3$$

$$x_i \geq 0, \; i=1,2,3$$

$$x^* = (3.15121$$
$$.216988$$
$$3.55217)$$

$$\lambda^* = (-1.22346$$
$$- .274937)$$

$$F(x^*) = 961.715$$

This example has been published in several comparisons (see Himmelblau, 1972).

Example 2(A)

Start (10,10,10)

	Iterations	Functions and Gradients
Penalty Method:		
$\rho = 10^2$	14	15
10^3	3	3
13^4	2	2
10	1	1
10^6	1	1
10^7	1	1
TOTAL	22	23

	Iterations	Functions end Gradients
Sequential Method, First-order multiplier update		
$\rho = 10^2$	14	15
	3	3
	1	1
TOTAL	18	19

Progressive Method, First-order multiplier estimates		
$\rho = 10^2$	14	15

Trajectory Method	11	11

Example 2(B)

Start (-5,-10,5)

NOTE: For all except the first minimization, the results for the penalty and sequential algorithms are the same as in Case 2(A).

Penalty Method		
$\rho = 10^2$	29	36
10^3	3	3
10^4	2	2
10^5	1	1
10^6	1	
10^7	1	1
TOTAL	37	44

	Iterations	Functions end Gradients
Sequential Method, First-order multiplier update		
$\rho = 10^2$	29	36
	3	3
	1	1
TOTAL	33	40
Progressive Method, First-order multiplier estimates $(\rho = 10^2)$	29	34
Trajectory Method	12	13

EXAMPLE 3:

This example was given by Powell (1969), and has since been quoted by a number of authors (see Fletcher, 1970; Glad, 1973).

The problem, as originally formulated, involves minimizing $F = x_1x_2x_3x_4x_5$; however, the resulting penalty function is unbounded below (see Section 2.3), so that the objective function is transformed, as follows:

$$\min e^{x_1x_2x_3x_4x}$$

subject to

$$c1 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$c_2 = x_2x_3 - 5x_4x_5 = 0$$

$$c_3 = x_1^3 + x_2^3 + 1 = 0$$

The solution usually given is:

$$x^* = (-1.71714$$
$$1.59571$$
$$1.82725$$
$$-.763643$$
$$-.763643)$$

$$\lambda^* = (-.0401627$$
$$.0379578$$
$$-.00522264)$$

$$F(x^*) = .0539499 .$$

This example is of some interest because it is normally used to demonstrate the success cf an augmented Lagrangian algorithm for a modest value of ρ. As discussed throughout Chapter 4, the penalty terra of an augmented Lagrangian method is designed to ad3 positive curvature in the range of $A(x^*)$, so that the Hessian of the augmented function,

$W(x^*) + \rho AA^T$, is positive definite.

For this exsmple, the eigenvalues of $G(x^*)$ are (.0494, .0576, . 7, .2701, and 1.575); the eigenvalues of $W(x^*)$ are (.0887, .1607, .1633, .2263, and 1.811). Therefore, the Hessian cf the Lagrangian function is positive definite at x^*, and the penalty term is not even needed - add positive curvature.

~~Example 3~~(A)

Start (-2,2,2,-1,-1)

Note that this starting point is very clcse to the solution.

	Iterations	Functions and Gradients
Penalty Method		
p = 1	7	7
10	3	3
1C	1	1
1C	1	1
10^4	1	1
TOTAL	13	13
Sequential Method, First-order multiplier update		
(ρ = 1)	7	7
	2	2
	1	1
	1	1
TOTAL	11	11
Progressive. Method, First-order multiplier estimates		
(ρ = 1)	8	8

	Iterations	Functions and Gradients
Trajectory Method	6	6

Example 3(B)

Start (-1,-1,-1,-1,-1)

Here, the initial penalty parameter was set to 10^2, making the first minimization more difficult.

Penalty Method		
$\rho = 10^2$	16	17
10^3	1	1
10^4	1	1
TOTAL	18	19
Sequential Method, First-order multiplier update		
$\rho = 10^2$	16	17
	1	1
TOTAL	17	18
Progressive Method, First-order multiplier estimate		
$(\rho = 100)$	18	19
Trajectory Method	6	7

Example 3(C)

Start (-2,-2,-2,-2,-2)

With this starting point, a second local minimum was found.

$$x^* = \begin{pmatrix} -.699050 \\ -.869952 \\ -2.78992 \\ -.696721 \\ -.696721 \end{pmatrix}$$

$$\lambda^* = (-.00437092$$
$$.131433$$
$$.310998)$$

$$F(x^*) = .438851$$

. This example illustrates the possible danger of relying on second-order multiplier estimates throughout, even if exact second derivatives are available. As mentioned in Section 4.8, the quadratic programming sub-problem on which the second-order multiplier estimates are based may have little meaning far from the solution. Some criteria must be developed to decide when the current iterate is "close enough" to rely on the second-order multiplier estimates. Note that the sequential method with a second-order multiplier update converges in one step after the initial minimization, so that the second-order estimate is of great value close to x^*.

	Iterations	Functions and Gradients
Penalty Method		
$\rho = 10^2$	59	82
1	4	4
10^4	3	3
10^5	1	1
10^6	1	1
10^7	1	1
TOTAL	69	92

	Iterations	Functions and Gradients
Sequential Method, First-order multiplier updates		
$(\rho = 10^2$	59	82
	3	3
TOTAL	63	86

	Iterations	Functions and Gradients
Progressive Method, First-order multiplier estimate	59	82
Sequential Method, Second-order multiplier update	59	82
	1	1
TOTAL	60	83
Progressive Method, Second-order multiplier estimate	62	102
	(not converged; final point wass not even close to solution)	
Trajectory Method	8	8

EXAMPLE 4:

$$\min\ (x_1-1)^2 + (x_1-x_2)^2 + (x_2-x_3)^3 + (x_3-x_4)^4 + (x_4-x_5)^4$$

subject to

$$c_{.} = x_{.}+x \qquad x_{n}^{3}-2-3\sqrt{2} = 0$$

$$c_2 = x_2-x_3^2 + x_4+2-2\sqrt{2} = 0$$

$$c_3 = x_1x_5-2 \qquad = 0$$

This problem is a modified version of an example given by Miele et al.(1971). Four different local minima have been determine6.

(1) $x^{*(1)}$ = (1.11663
1.22044
1.53779
1.97277
1.79110)

$\lambda^{*(1)}$ = (.0641296
.353202
-.0214797)

$F(x^{*(1)})$ = .0293183.

The eigenvalues of $W(x^{*(1)})$ are (-2.49, .208, .605, 4.10, 5.09), so that $W(x^*)$ is not positive definite. The eigenvalues of $W + 10\ AA^T$ are (.538, 2.10, 40.3, 50.8, 646.0), so that p = 10 is sufficient to assure positive curvature at $x^{*(1)}$.

$$(2) \quad x^{*(2)} = \begin{pmatrix} -2.79087 \\ 3.00414 \\ .205371 \\ 3.87474 \\ -.716623 \end{pmatrix}$$

$$\lambda^{*(2)} = \begin{pmatrix} 92.2563 \\ 584.778 \\ 138.722 \end{pmatrix}$$

$$F(x^{*(2)}) = 607.036.$$

The eigenvalues of $W(x^{*(2)})$ are (-202.1, -88.25, 139.5, 585.6, 1233.0), so that $W(x^*)$ has two negative eigenvalues. The eigenvalues of $W + 10\ AA^T$ are (-54.05, 132.7, 208.9, 619.1, 1237.1), so that $\rho = 10$ is not sufficient to make the augmented Hessian positive definite at $x^{*(2)}$. The eigenvalues of $W + 100\ AA^T$ are (43.81, 372.1, 1103.0, 1287.0, 3621.0), so that $\rho = 100$ is sufficiently large.

$$(3) \quad x^{*(3)} = \begin{pmatrix} -1.27305 \\ 2.41035 \\ 1.19486 \\ -.154239 \\ -1.57103 \end{pmatrix}$$

$$\lambda^{*(3)} = \begin{pmatrix} 2.12527 \\ 1.55380 \\ 8.93568 \end{pmatrix}$$

$$F(x^{*(3)}) = 27.8719$$

The eigenvalues of $W(x^{*(3)})$ are (-6.75, 2.68, 7.23, 24.2, 68.7). The eigenvalues of $W + 10\ AA^T$ are (8.09, 28.6, 54.1, 110.0, 440.0).

(4) $x^{*(4)} = (-.703393$

2.63570

$-.0963618$

-1.79799

$-2.84336)$

$\lambda^{*(4)} = (\ 8.38670$

-15.1290

$6.49642)$

$F(x^{*(4)}) = 44.0221.$

The eigenvalues of $W(x^{*(4)})$ are (-14.1, .818, 8.26, 21.0, 76.3).

The eigenvalues of $W + AA^T$ are (1.47, 17.7, 75.6, 88.6, 303.0)

Example 4(A)

Start (1,i,1,1,1)

With this starting point, all methods converged to the local minimum $x^{*(1)}$.

	Iterations	Functions and Gradients
Penalty Method		
$p = 10^2$	14	17
10	3	3
10^4	2	2
1C	1	1
10^6	1	1
10^7	1	1
TOTAL	22	25
Sequential Method, First-order multiplier update		
$(p = 10^2)$	14	17
	2	2
	1	1
	1	1
TOTAL	18	21
Progressive Method, First-order multiplier estimate		
$(\rho = 10^2)$	14	17
Trajectory Method	11	19

Example 4(B)

Start (2,2,2,2,2)

. All methods converged to $x^{*(1)}$.

	Iterations	Functions and Gradients
Penalty Method.		
$\rho = 10^2$	15	18
10^3	3	3
10^4	2	2
10^5	1	1
10^6	1	1
10^7	1	1
TOTAL	23	26
Sequential Method, First-order multiplier update		
$(\rho = 10^2)$	15	18
	2	2
	1	1
	1	1
TOTAL	19	22
Progressive Method, First-order multiplier estimate $(\rho = 10^2)$	15	17
Trajectory Method	11	12

Example 4(C)

Start $(-1,3,-\frac{1}{2},-2,-3)$

Here, the progressive and trajectory algorithms converged to $x^{*(4)}$, while the penalty and sequential algorithms converged to $x^{*(3)}$. The local advantage of the second-order multiplier update is displayed for the sequential method.

	Iterations	Functions and Gradients
Penalty Method		
$p = 10^2$	14	20
10^3	4	4
10_4	3	3
10^5	3	3
10^6	1	1
10^7	1	1
10^8	1	1
TOTAL	27	33
Sequential Method, First-order multiplier update		
$(p = 10^2)$	14	20
	3	3
	2	2
	1	1
TOTAL	20	26
Sequential Method, Second-order multiplier update	14	20
	2	2
TOTAL	16	22

	Iteration	Functions and Gradients
Progressive Method, First-order multiplier estimate (converged to $x^{*(4)}$)	11	12
Trajectory Method (converged to $x^{*(4)}$)	11	11

Example 4(D)

Start (-1,2,1,-2,-2)

. With this starting point, the progressive method converged to $x^{*(4)}$, while the others converged to $x^{*(3)}$. Again, the improved local convergence of the sequential method with a second-order multiplier update is seen.

	Iteration	Functions and Gradients
Penalty Method		
$\rho = 10^2$	12	17
10^3	4	4
10^4	3	3
10^5	3	3
10^6	1	1
10^7	1	1
10^8	1	1
TOTAL	25	30
Sequential Method, First-order multiplier update		
$(\rho = 10^2)$	12	17
	3	3
	2	2
	1	1
TOTAL	18	23
Sequential Method, Second-order multiplier update		
$(\rho = 10^2)$	12	17
	[illegible]	
	2	2
TOTAL	14	18

	Iteration	Functions and Gradients
Trajectory Method	11	17
Progressive Method, First-order multiplier estimate (converged to $x^{*(4)}$)	13	20

Example 4(E)

Start (-2,-2,-2,-2,-2)

With this starting point, all methods converged to $x^{*(2)}$, except for the trajectory algorithm which converged to $x^{*(4)}$. The sequential algorithm with first-order multiplier updates increased the penalty parameter to 10^4, whereas ρ remained at 10^2 for the sequential method with second-order multiplier updates.

This example illustrates the impediment to convergence of a progressive algorithm posed by the accuracy of the multiplier estimate. The progressive method with first-order multiplier estimates had reduced $||c||$ to 5×10^{-3} by iteration 33; but the next 24 iterations display slow linear convergence to x^*. On the other hand, the progressive algorithm with second-order multiplier estimates converges rapidly.

		Iterations	Functions and Gradients
Penalty Method			
	$p = 10^2$	11	15
	10^3	6	6
	104	4	4
	10^5	3	3
	106	2	2
	10^7	1	1
	10^8	1	1
	10^9	1	1
TOTAL		29	33

	Iterations	Gradients
Sequential Method, First-order multiplier update		
($\rho = 10^2$)	11	15
	4	4
($\rho = 10^3$)	4	4
	3	3
($\rho = 10^4$)	3'	3
	2	2
	1	1
	1	1
TOTAL	29	33
Sequential Method, Second-order multiplier update	11	15
	3	3
	2	2
TOTAL	16	20
Progressive Method, First-order multiplier estimates	57	61
	($\Vert\varepsilon\Vert \equiv 5$ x10^{-3} at iteration 33)	
Progressive Method, Second-order multiplier estimates	11	13
Trajectory Method (converges to $x^{*(4)}$)	22	36

6.3.2 Feasible Methods

The unconstrained sub-problem for the barrier functicn method terminated when $||\text{grad}|| \leq 10^{-5}$; the overall convergence criteria was that $||\hat{c}|| \leq 10^{-5}$ for the "active" constraints. The evaluations of the objective function and constraints are listed separately, because the objective function is evaluated only at feasible points. The linear search tolerance, η, was 0.2.

EXAMPLE 5:

$$\min\ x_1 x_2^2$$

subject to

$$c_1 = -x_1^2 - x_2^2 + 2.0 \geq 0$$

$$x^* = (-.816497$$
$$-1.15470)$$

$$\lambda^* = (.816497)$$

This example is the same as example 1, and was selected so that the convergence paths could be plotted. The results illustrate how the usual barrier function algorithm can be "caught" near the boundary of the Teasible region, far from x*, and be forced to take many small steps close to the boundary.

Example 5(A)

Start (-1.0, -.7)

Figure 6.2 displays the steps taken by the barrier function method and the trajectory method.

	Iterations	Obj. Functions and Gradients	Constraints and Gradients
Barrier Function Method			
$r = 10^{-2}$	19	42	55
10^{-3}	3	3	3
10^{-4}	3	8	8
10^{-5}	2	2	2
10^{-6}	1	1	1
TOTAL	28	56	69
Trajectory Method	4	6	7

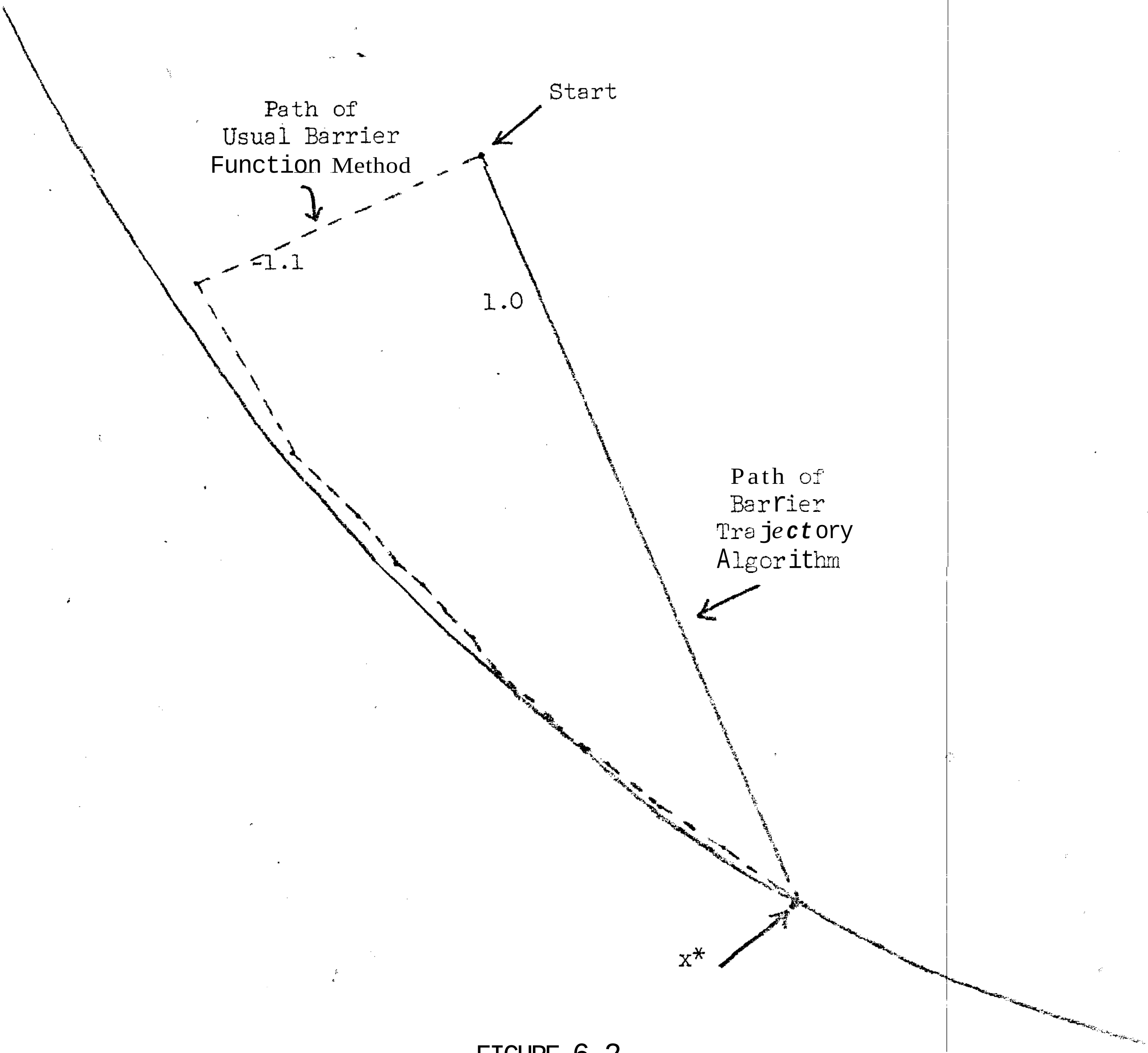

Start
Path of
Usual Barrier
Function Method
1.1
1.0
Path of
Barrier
Trajectory
Algorithm
x*

FIGURE 6.2

Example 5(B)

Start (-.8,- .6)

Figure 6.3 displays the paths taken to x* for this starting point.

		Obj.Functions	Con4traints
	Iterations	and Gradients	and Gradients
Barrier Function Method			
$r = 10^{-2}$	19	39	52
10^{-3}	2	3	3
	2	8	8
10^{-5}	2	2	2
TOTAL	25	52	65
Trajectory Method	4	5	7

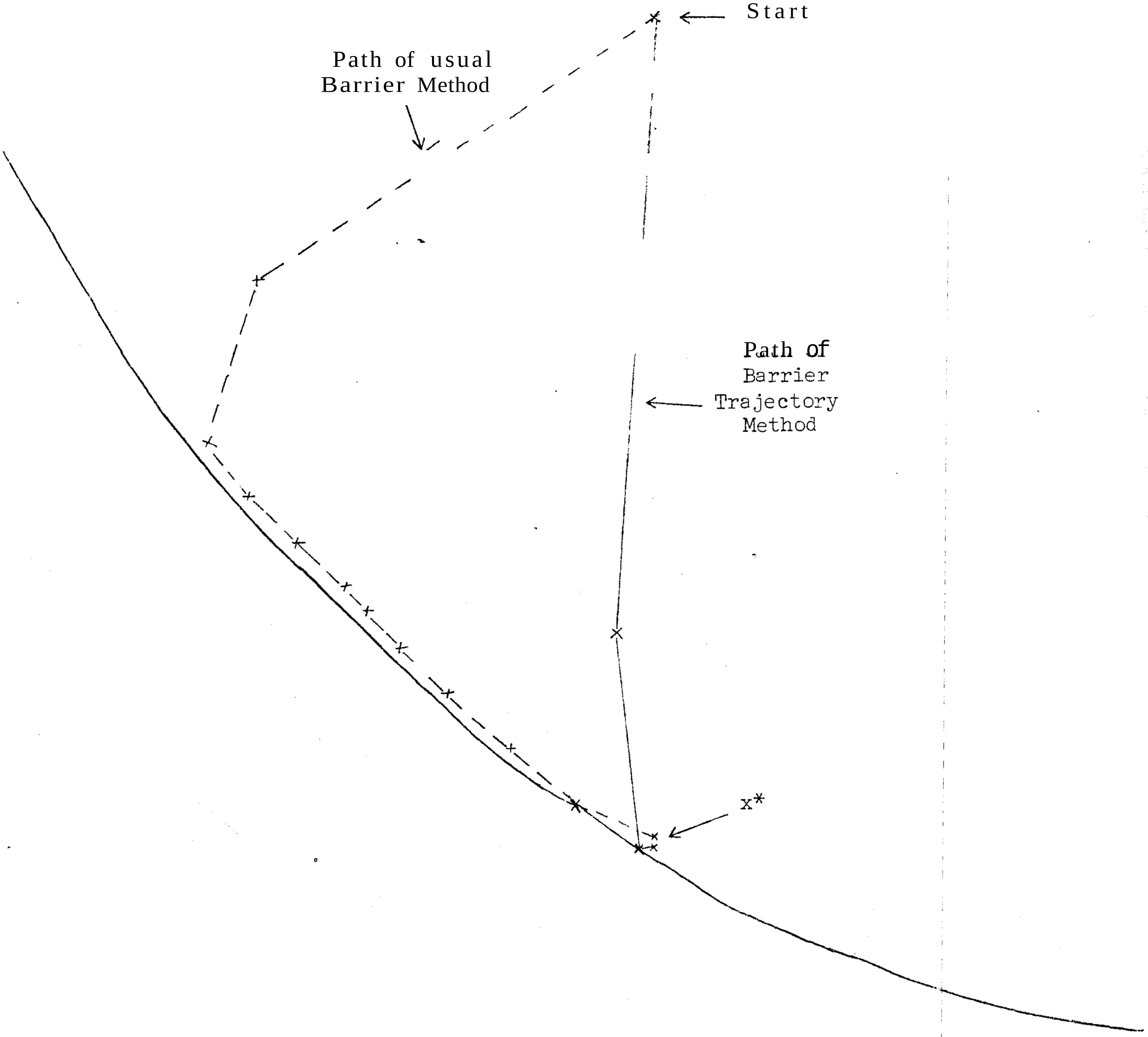

Start
Path of usual Barrier Method
Path of Barrier Trajectory Method
x*

FIGURE 6.3

EXAMPLE 6:

$$\min x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to

$$c_1 = -x_1^2 - x_2^2 - x_1^2 - x4^2 - x_1 + x_2 \geq 0$$

$$c2 = -x1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 \geq 0$$

$$c_3 = -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 \geq 0$$

This example is due to Rosen and Suzuki (1965).

$$x^* = (0.0$$
$$1.0$$
$$2.0$$
$$-1.0)$$

Constraints c1 and c3 are active, so that $I = \{1,3\}$; $\lambda^* = (1.0,2.0)$.

Start (0,0,0,0)	Iteration	Obj.Function and Gradients	Constraints and Gradients
Barrier Function Method			
r = 1			
10^{-1}			
10^{-2}			
10^{-3}			
TOTAL			
Trajectory Method			

EXAMPLE 7:

$$\min e^{x_1 x_2 x_3 x_4 x_5}$$

subject to

$$c1 = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_5^2 + 10 \geq 0$$

$$c2 = x_2 x_3 - 5 x_4 x_5 \geq 0$$

$$c_3 = -x1^3 - x_2^3 - 1 \geq 0$$

The problem here is the same as Example 3, the Powell exponential, with the constraints converted to inequalities to correspond to the signs of the multipliers. The starting point is chosen very close to the solution in order to assure that all three constraints are active. It should be noted that the exponential transformation is not necessary for the feasible methods, and was retained in order to facilitate comparison among the algorithms.

Start (-1.74,1.61,1.81,-.7,-.7)

	Iterations	Obj.Function and Gradient	Constraints and Gradients
Barrier Function Method			
$r = 10^{-2}$	7	9	9
10^{-3}	6	9	9
10^{-4}	5	8	8
10^{-5}	4	6	6
10^{-6}	4	6	6
10^{-7}	4	6	6
TOTAL	30	44	44
Trajectory Method	9	10	11

EXAMPLE *8:*

This example is called the "hexagon" problem (see Murray, 1969a). It involves maximizing the area of a hexagon, no two of whose vertices are more than one unit apart. The constraints are derived from this requirement, and from the problem geometry, which is represented in Figure 6.4.

Hexagon Problem:

$$\min -\frac{1}{2}(x_2x_6 - x_1x_7 + x_3x_7 + x_5x_8 - x_4x_9 - x_3x_8)$$

subject to

$$c_1 = 1 - x_1^2 - x_2^2 \geq 0$$
$$c_2 = 1 - (x_2-x_1)^2 - (x_7-x_6)^2 \geq 0$$
$$c_3 = 1 - (x_3-x_1)^2 - x_6^2 \geq 0$$
$$c_4 = 1 - (x_1-x_4)^2 - (x_6-x_8)^2 \geq 0$$
$$c_5 = 1 - (x_1-x_5)^2 - (x_6-x_9)^2 \geq 0$$
$$c_6 = 1 - x_2^2 - x_7^2 \geq 0$$
$$c_7 = 1 - (x_3-x_2)^2 - x_7^2 \geq 0$$
$$c_8 = 1 - (x_4-x_2)2 - (x_8-x_7)2 \geq 0$$
$$c_9 = 1 - (x_2-x_5)^2 - (x_7-x_9)^2 \geq 0$$
$$c_{10} = 1 - x_3^2 \geq 0$$
$$c_{11} = 1 - (x_4-x_3)^2 - x_8^2 \geq 0$$
$$c_{12} = 1 - (x_5-x_3)^2 - x_9^2 \geq 0$$
$$c_{13} = 1 - x_4^2 - x_8^2 \geq 0$$
$$c_{14} = 1 - (x_4-x_5)^2 - (x_9-x_8)^2 \geq 0$$
$$c_{15} = 1 - x_5^2 - x_9^2 \geq 0$$

$$c_{16} = x_1 \geq 0$$
$$c_{17} = x_6 \geq 0$$
$$c_{18} = x_5 \geq 0$$
$$c_{19} = x_2 - x_1 \geq 0$$
$$c_{20} = x_7 \geq 0$$

$$c_{21} = x_3 - x_2 \geq 0$$
$$c_{22} = x_4 - x_5 \geq 0$$
$$c_{23} = -x_8 \geq 0$$
$$c_{24} = -x_9 \geq 0$$
$$c_{25} = x_3 - x_4 \geq 0$$

Hexagon Geometry

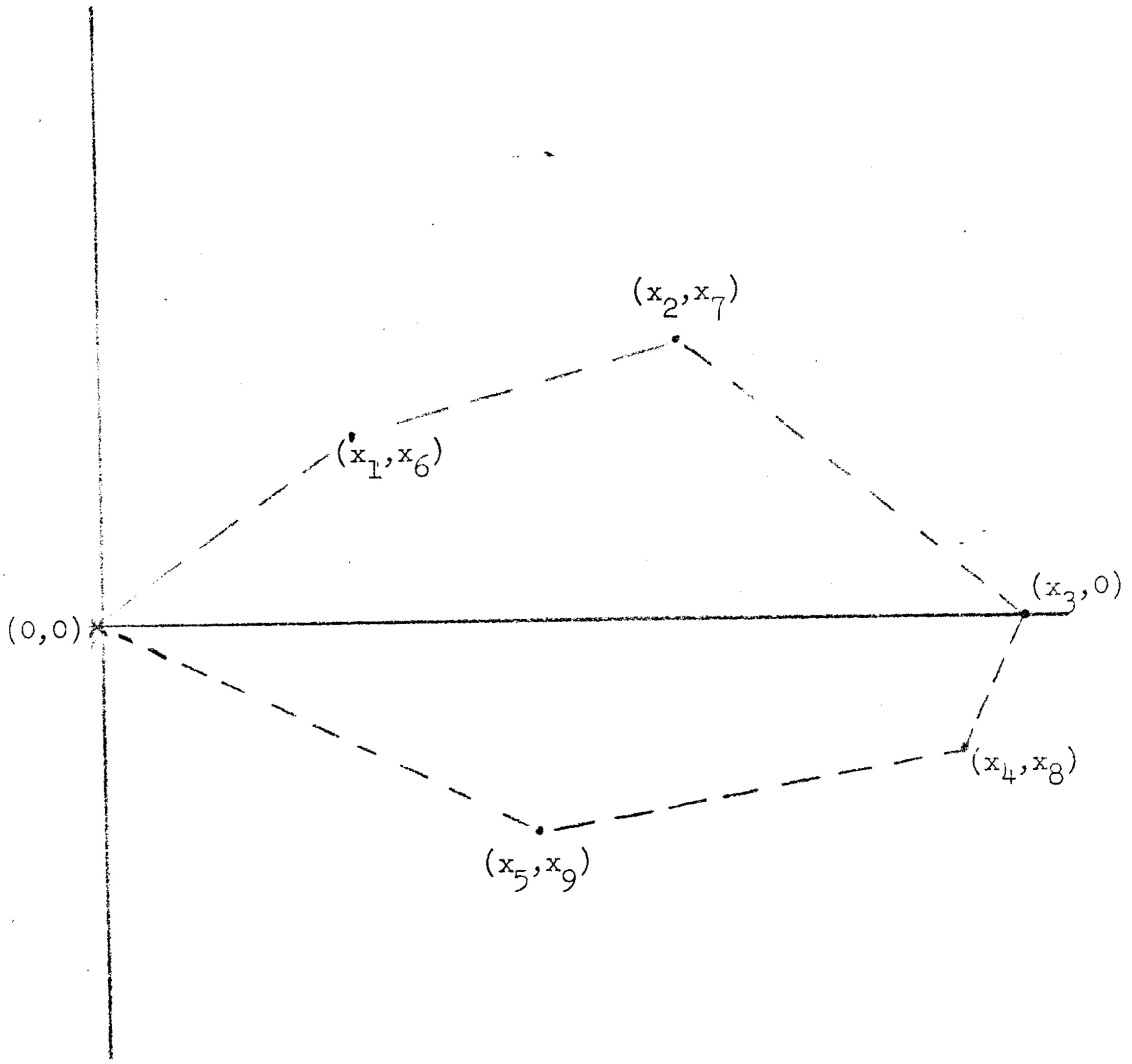


FIGURE 6.4

$$x^* = (\ .257355$$
$$.580129$$
$$1.00000$$
$$.794057$$
$$.156229$$
$$.235929$$
$$.396006$$
$$-\ .607843$$
$$-\ .536703)\ .$$

The active set $I = \{4, 8, 9, 10, 12, 13\}$,

and $\lambda^* = (\ .171886$
$$.0415920$$
$$.160131$$
$$.160131$$
$$.099649$$
$$.0415920)$$

$F(x^*) = -.674981$

(max. area = .674981)

Figure 6.5 displays the optimum configuration found (there are other equivalent arrangements that produce the same maximum ares).

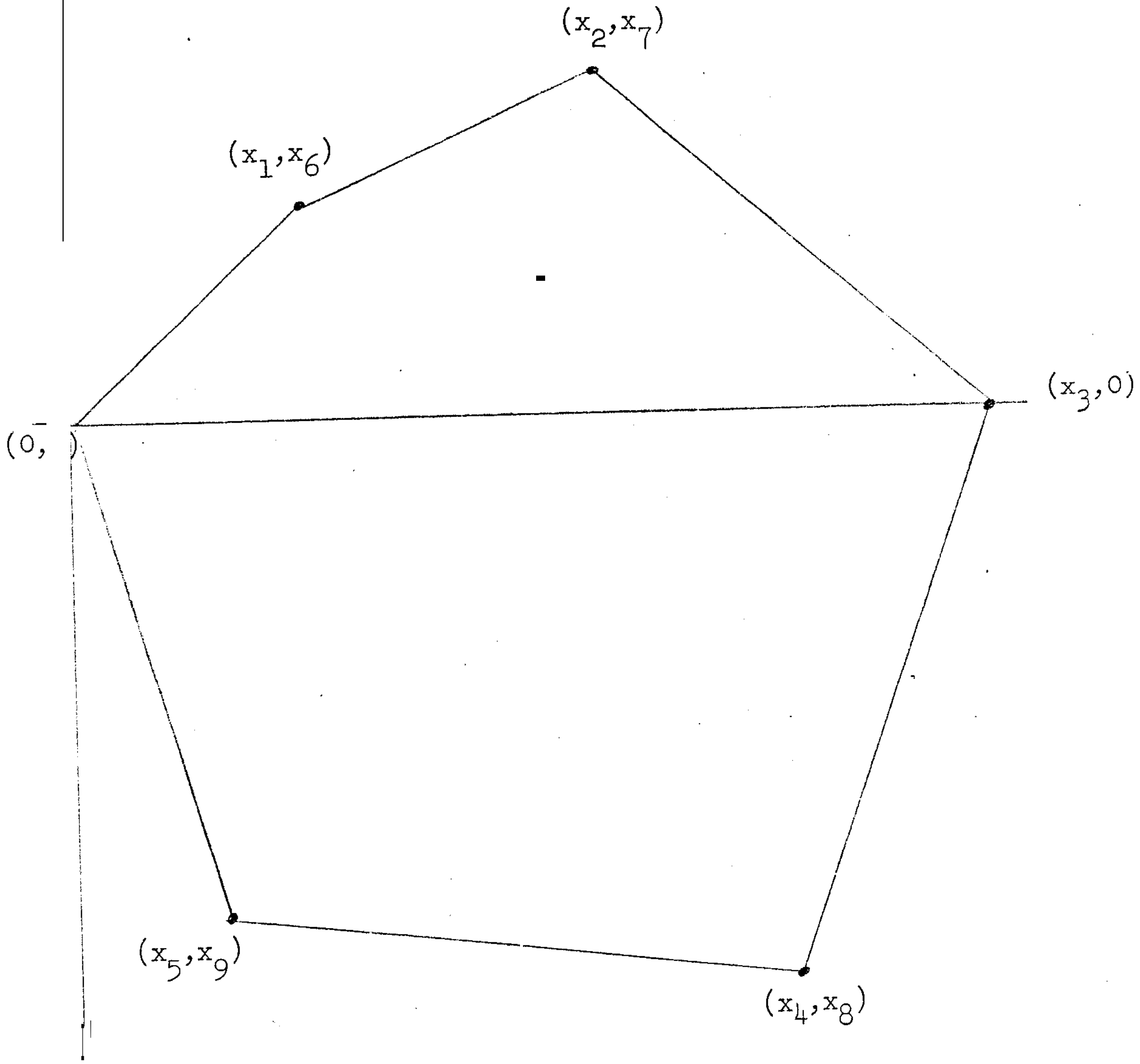


FIGURE 6.5

Start $(\frac{1}{10}, \frac{1}{8}, \frac{1}{6}, \frac{1}{7}, \frac{1}{9}, \frac{1}{5}, \frac{1}{4}, -\frac{1}{5}, -\frac{1}{4})$

	Iterations	Obj.Function and Gradient	Constraints and Gradients
Barrier Function Method			
$r = 10^{-2}$	15	27	36
10^{-3}	10	16	17
10^{-4}	7	9	10
10^{-5}	8	13	14
10^{-6}	8	14	16
10^{-7}	5	9	9
TOTAL	53	88	102

EXAMPLE 9:

$$\min \quad 10x_1x_4 - 6x_3x_2^2 + x_2x_1^3 + 9\sin(x_5-x_3) + x_5^4x_4^2x_2^3$$

subject to

$$\begin{aligned}
c_1 &= -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_5^2 + 20 \geq 0\\
c_2 &= x_1^2x_3 + x_5x_4 + 2 \qquad\qquad \geq 0\\
c_3 &= x_2^2x_4 + 10x_1x_5 - 5 \qquad\qquad \geq 0
\end{aligned}$$

This example provides ample illustration of the need to try different starting points. The behavior of the algorithms alters dramatically in this case, depending on the initial point.

The results for this example were obtained with an earlier version of the linear search, where a less efficient procedure was used to obtain the next step if a constraint is violated during the linear search; the number of constraint evaluations could no doubt be reduced with the improved procedure described in Section 3.3.

Example 9(A)

Start (1,1,1,1,1)

Both methods converge to the following local minimum:

$$\begin{aligned}
x^{*(1)} = (&-.0814522\\
&3.69238\\
&2.48741\\
&.377134\\
&.173983)
\end{aligned}$$

Constraints c_1 and c_3 are active, so that $I = \{1,3\}$, and $\lambda_*^{(1)} = (15.2198, .784830)$.

	Iterations	Obj.Function and Gradients	Constraints and Gradients
Barrier Function Method			
$r = 1$	40	47	75
10^{-}	12	14	16
10^{-}	8	12	13
10^{-3}	5	9	10
TOTAL	65	82	114
Trajectory Method	20	34	49

Example 9(B)

Start (1.091,-3.174,1.214,-1.614,2.134)

This starting point is close to the constraint boundary, and the usual barrier algorithm exhibited the phenomenon of being "caught" nea the boundarv for hundreds of iterations. Both methods converge to the following local minimum:

$$x^{*(2)} = (1.47963 \\ -2.63661 \\ 1.05467 \\ -1.61151 \\ 2.67388)$$

Constraints 1 and 2 are active, so that $I = \{1,2\}$, and $\lambda^{*(2)} = (531.963, 23.3609)$.

	Iterations	Obj .Function and Gradient	Constraints and Gradients
Barrier Function Method			
r = 1	306	619	819
10^{-1}	4	7	7
	4	6	6
10^{-3}	3	5	5
TOTAL	317	627	837
Trajectory Method	104	343	439

Note: When the trajectory algorithm was started at $x^*(1)$, it converged to x^* in 6 steps, with 6 function and constraint evaluations, compared to 11 steps, 18 function and constraint evaluations, for the usual barrier function methods.

CHAPTER 7

CONCLUSIONS

The test results presented in Section 6.3 illustrate numerically most' of the expected theoretical properties of the various algorithms.

Augmented Lagrangian Methods

Some notable aspects of computation with augmented Lagrangian algorithms are:

(1) The local rate of convergence of augmented Lagrangian methods is restricted to the rate of convergence of the multiplier estimates (see Example 4(E));

(2) Even "second-order" multiplier estimates may be unreliable if the starting point is away from the solution (see Example 3(C));

(3) In general, the progressive augmented Lagrangian methods display improved efficiency compared with the sequential methods.

The sequential augmented Lagrangian algorithms have the advantage that the unconstrained sub-problem can be solved by straightforward application of an unconstrained method; however, in many cases the sequential methods converged no more rapidly than the penalty function method, because the value of the penalty parameter required for convergence was not large enough to induce serious ill-conditioning of the Hessian of the penalty function. With the progressive methods, the frequent updating of multiplier estimates and/or penalty parameter makes direct use of' an existing unconstrained method difficult.

For any of the augmented Lagrangian methods, the rate of convergence of the multiplier estimates is crucial (for details, see Section 4.7).

Since second derivatives of the problem functions will not in general be available, an augmented Lagrangian method cannot rely on being able to compute second-order multiplier estimates. Some recent work has focused on developing multiplier updates to accompany quasi-Newton methods applied to progressive augmented Lagrangian algorithms; however, the validity of the quasi-Newton Hessian approximation required for such updates is questionable, especially since the function to be approximated varies from iteration to iteration.

Trajectory Algorithms

The trajectory methods were consistently successful, and compare favorably - in some cases very favorably - with the best of the other methods. The theoretical derivation and numerical results indicate the following strengths of the trajectory methods as general algorithm:

(1) Their success does not depend on being in a close neighborhood of the solution, since they are based on the properties of penalty and barrier functions;

(2) Their local rate of convergence is not restricted to the rate of convergence of the Lagrange multiplier estimates;

(3) A serious difficulty of a general barrier function algorithm is that many iterations can be essentially wasted in small steps,close to the boundary of the feasible region. The derivation of the barrier trajectory algorithm indicates that it should not be subject to this tendency, and this expectation has been verified in the examples tested.

(4) Because the minimization sub-problem at each iteration of a trajectory method is solved in a reduced subspace, the use of finite differences to approximate the projected

Hessian becomes reasonable for most problems, where the number of active constraints is a significant proportion of the number of variables. Hence, a practical second-order rate of convergence can be attained even when only gradients of the problem functions are available.

BIBLIOGRAPHY

Abadie, J. and Carpentier, J. (1969). "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in .Optimization (R. Fletcher, ed.), pp. 37-49, Academic Press, London and New York.

Biggs, M.C. (1972). "Constrained Minimization using Recursive Eqcality Quadratic Programming," in Numerical Methods for Non-Linear Optimization (F.A. Lootsma; ed.), pp. 411-428, Academic Press, London ant? New York.

Biggs, M.C. (1974). The Development of a Class of Constrained Optimization Algorithms and Their Application to the Problem of Electric Power Scheduling, Ph.D. Thesis, University of London.

Brent, R.P. (1973). Algorithms for Minimization without Derivatives, Prentice-Hall, Englewcod Cliffs, New Jersey.

Broyden, C.G. (1972). "Quasi-Newton Methods," in Numerical Methods for Unconstrained Optimization (W. Murray, ed.), pp. 87-106, Academic Press, London and New York.

Buys, J.D. (1972). Dual Algorithms for Constrained Optimization Problems, Ph.D. Thesis, University of Leiden.

Carroll, C.W. (1961). "The Created Response Surface Technique for OFtimizing Nonlinear Restrained Systems,.. Operations Research, vol. 9, pp. 169-184.

Cook, A.J. and Shustek, L.J. (1975). A User's Guide to MORTRAN2, CGTM No. 165, Computation Research Group, Stanford Linear Accelerator Center.

Courant, R. (1943). "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," bull. Am. Math. Soc., vol. 49, pp. 1-23.

Davidon, W.C. (1959). Variable Metric: Method for Minimization, Argonne National Lab. Report No. ANL-5990.

Dennis, J. and Mei, H. (1975). An Unconstrained Optimization Algorithm which uses Function and Gradient Values, Report TR-75-246, Department of Computer Science, Cornell University.

Fiacco, A.V. and McCormick, G.P. (1968). Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley and Sons, New York.

Fletcher, R. (1970). "A Class of Methods for Nonlinear Programming with Termination and Convergence Properties," in Integer and Nonlinear Programing (J. Abadie, ed.), pp. 157-175, North-Holland, Amsterdam.

Fletcher, R. (1973a). An Ideal Penalty Function for Constrained Optimization, Report CSS2, A.E.R.E. Harwell.

Fletcher, R. (1973b). An Exact Penalty Function for Nonlinear Programming with Inequalities, Mathematical Programing, vol. 5, pp. 129-150.

Fletcher, R. (1974). "Methods Related to Lagrangian Functions," in Numerical Methods for Constrained Optimization (P.E. Gill and W. Murray, eds.), pp. 219-240, Academic Press, London and New York.

Fletcher, R. and Lill, S. (1970). "A Class of Methods for Nonlinear Programming II: Computational Experience," in Nonlinear Programming (J.B. Rosen, O.L. Mangasarian and K. Ritter, eds.), pp. 67-92, Academic Press, London and New York.

Fletcher, R. and McCann, A.P. (1969). "Acceleration Techniques for Nonlinear Programming," in Optimization (R. Fletcher, ed.), pp. 203-213, Academic Fress, London and New York.

Fletcher, R. and Powell, M.J.D. (1963). A Rapidly Convergent Descent Method for Minimization, Computer Journal, vol. 6, pp. 163-168.

Frisch, K.R. (1955). The Logarithmic Potential Method of Convex Programming, Memorandum, May 13, 1955, University Institute of Economics, Oslo.

Gill, P.E., Golub, G.H., Murray, W. and Saunders, M.A. (1974). Methods for Modifying Matrix Factorizations, Mathematics of Computation, vol. 28, pp. 505-535.

Gill, P.E. and Murray, W. (1972a). Two Methods for the Solution of Linearly Constrained and Unconstrained Optimization Problems, Report NAC-25, National Physical Laboratory.

Gill, P.E. and Murray, W. (1972b). Quasi-Newton Methods for Unconstrained Optimization, Journal Inst. Math. Appl., vol. 9, pp. 91-108.

Gill, P.E. and Murray, W. (1974a). Safeguarded Steplength Algorithms for Optimization using Descent Methods, Report NAC-37, National Physical Laboratory.

Gill, P.E. and Murray, W. (1974b). "Newton-type Methods for Linearly Constrained Optimization," in Numerical Methods for Constrained Optimization (P.E. Gill and W. Murray, eds.), pp. 29-66, Academic Press, London and New York.

Glad, T. (1973). Lagrange Multiplier Methods for Minimization under Equality Constraints, Report 7323, Lund Institute of Technology.

Golub, G.H. and Pereyra, V. (1973). The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems whose Variables Separate, SIAM Journal Num. Analysis, vol. 10, pp. 413-432.

Haarhoff, P.C. and Buys, J.D. (1970). A New Method for the Optimization of a Nonlinear Function subject to Nonlinear Constraints, Computer Journal, vol. 13, pp. 178-184.

Hestenes, M.R. (1969). Multiplier and Gradient Methods, Journal Opt. Theory Appl., vol. 4, pp. 303-320.

Himmelblau, D. (1972). Applied Nonlinear Programming, McGraw-Hill, New York.

Kuhn, H.W. and Tucker, A.W. (1951). "Nonlinear Programming," in Proceedings of the Second Berkeley Symposium on Mathematical Statistics 'and Probability (J. Neyman, ed.), pp. 481-493, University of California Press-.

Lasdon, L.S., Fox, R.L., and Ratner, M.W. (1973). An Efficient One-Dimensional Search Procedure for Barrier Functions, Mathematical Programming, vol. 4, pp. 279-295.

Lasdon, L.S., Waren, A.D., Jain, A. and Ratner, M.W. (1975). Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Optimization, Tech. Memo 353, Department of Operations Research, Case Western Reserve University.

Levenberg, K. (1944). A Method for the Solution of Certain Problems in Least-Squares, Quart. Appl. Math., vol. 2, pp. 164-168.

Lill, S.A. (1972). "Generalization of an Exact Method for Solving Equality Constrained Problems to Deal with Inequality Constraints," in Numerical Methods for Non-Linear Optimization (F. Lootsma, ed.), pp. 383-394, Academic Press, London and New York.

Lootsma, F.A. (1969). Hessian Matrices of Penalty Functions for Solving Constrained Optimization Problems, Phillips Res. Report 24, pp. 322-331.

Lyness, J. and Kaganove, J. (1976). A Technique for Comparing Automatic Quadrature Routines, Argonne National Laboratory.

Marquardt, I?. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters, SIAM Journal Appl. Math., vol. 11, pp. 431-441.

Miele, A. Cragg, E.E., Iyer, R.R. and Levy, A.V. (1971). Use of the Augmented Penalty Function in Mathematical Programming Problems, Journal Opt. Theory Appl., vol. 8, pp. 115-153.

Murray, W. (1969a). Constrained Optimization, Report MA79, National Physical Laboratory.

Murray, W. (1969b). "An Algorithm for Constrained Minimization," in Optimization (R. Fletcher, ed.), pp. 247-258, Academic Press, London and New York..

Murray, W. (1971). Analytical Expressions for the Eigenvalues and Eigenvectors of the Hessian Matrices of Barrier and Penalty Functions, Journal Opt. Theory Appl., vol. 7, pp. 189-196.

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

Ostrowski, A. (1966). Solutions of Equations and Systems of Equations, Academic Press, New York and London.

Peters, G. and Wilkinson, J.H. (1970). The Least-Squares Problem and Pseudo-Inverses, Computer Journal, vol. 13, pp. 309-316.

Powell, M.J .D. (1969). "A Method for Nonlinear Constraints in Minimization Problems," in Optimization (R. Fletcher, ed.), pp. 283-298, Academic Press, London and New York.

Powell, M.J .D (1970). "A New Algorithm for Unconstrained Optimization," in Nonlinear Programming (J.B. Rosen, O.L. Mangasarian, and K. Ritter, eds.), pp. 31-65, Academic Press, London and New York.

Powell, M.J .D. (1972). "Problems Related to Unconstrained Optimization," in Numerical Methods for Unconstrained Optimization (W. Murray, ed.), pp. 29-55, Academic Press, London and New York.

Robinson, S.K. (1972). A Quadratically Convergent Algorithm for General Nonlinear Programing Problems, Mathematical Programning, vol. 3, pp. 145-156.

Robinson, S.M. (1973). Perturbed Kuhn-Tucker Points and Rates of Convergence for a Class of Nonlinear Programming Algorithms, M.R.C. Tech. Report 1298, University of Wisconsin.

Rockafellar, R.T. (1970). "New Applications of Duality in Nonlinear Programming," presented at Seventh International Symposium on Mathematical Programming, The Hague.

Rockafellar, R.T. (1973a). The Multiplier Method of Hestenes and Powell Applied to Convex Programming, Journal Opt. Theory Appl., vol. 12, pp. 555-562.

Rockafellar, R.T. (1973b). A Dual Approach to Solving Nonlinear Programing Problems by Unconstrained Optimization, Mathematical Programming, vol. 5, pp. 354-373.

Rosen, J.B. (1961). The Gradient Projection Method for Nonlinear Programming. Part 11: Nonlinear Constraints, SUM Journal Appl. Math., vol. 9, pp. 514-532.

Rosen, J.B. and Kreuser, J. (1972). "A Gradient Projection Algorithm for Nonlinear Constraints," in Numerical Methods for Non-linear Optimization (F.A. Lootsma, ed.), pp. 297-300, Academic Press, London and New York.

Rosen, J.B. and Suzuki, S. (1965). Construction of Nonlinear Programming Test Problems, Comm. ACM, vol. 8, p. 113.

Ryan, D.M. (1974). "Penalty and Barrier Functions," in Numerical Methods for Constrained Optimization (P.E. Gill and W. Murray, eds.), pp. 175-190, Academic Press, Londcn and New York.

Sargent, R.W. and Murtagh, B.A.H. (1973). Projection Methods for Nonlinear Programing, Mathematical Programming, vol. 4, pp. 245-268.

Tapia, R.A. (1975). Quasi-Newton Methods for Constrained Optimization, Rice University.

Wilson, R.B. (1.963). A Simplicial Algorithm for Concave Programming, Ph.D. Thesis, Graduate School of Business Administration, Harvard University.