DOE/ER/05046-16

AUGMENTED LAGRANGIAN METHODS FOR CONSTRAINED OPTIMIZATION: THE ROLE OF THE PENALTY CONSTANT

CONF-7906140--2

R. A. TAPIA

Department of Mathematical Sciences
Rice University, Houston, Texas

MASTER

Presented at the International Conference on Nonlinear Optimization and Applications, L'Aquila, Italy, June 1979.

DISCLAIMER

Augmented Lagrangian Methods for Constrained Optimization: The Role of the Penalty Constant[1]

R. A. Tapia[2]

In recent years there has been considerable research activity in the area of penalty function and augmented Lagrangian methods for constrained optimization. In this paper we review the role that the penalty constant plays with respect to local convergence and rate of convergence. We see that as the emphasis has changed from the penalty function methods to the multiplier methods, and lately to the quasi-Newton methods, there has been a corresponding decrease in the importance of the penalty constant. Specifically, in the penalty function method one obtains local convergence if and only if the penalty constant becomes infinite. It is possible to obtain local convergence in the multiplier method for a fixed penalty constant, provided that this constant is sufficiently large. However, one obtains superlinear convergence if and only if the penalty constant becomes infinite. Finally, the quasi-Newton methods are locally superlinearly convergent for fixed values of the penalty constant and actually the most natural formulation gives an algorithm which is independent of the penalty constant.

---

1. This work was supported by the United States Department of Energy under Contract EY-76-S-05-5046.

2. Department of Mathematical Sciences, Rice University, Houston, Texas 77001.

I. Introduction and Preliminaries.

Much of the background material on multiplier methods and quasi-Newton methods for constrained optimization is taken from Tapia (1977) and Tapia (1978). In describing an iterative procedure we often will not use subscripts or superscripts. Instead, we place a bar over quantities which correspond to the successive iterate, e.g., if x denotes the present iterate, then $\bar{x}$ denotes the next iterate. If $F:R^n \to R^m$, then F_i will denote the i-th component function, $F'(x)$ will denote the Jacobian matrix of F at x and $\nabla F(x)$ denotes $F'(x)^T$ (transpose). In the case that $m=1$, we see that $\nabla F(x)$ will be the gradient of F at x and $\nabla^2 F(x)$ $(=\nabla(\nabla F(x)))$ will be the Hessian matrix of F at x. We will also have occasion to consider F as a function of two vector variables, say $F(x,\lambda)$. We use the subscript x or λ to denote the partial derivative with respect to x or λ. No subscripts, e.g., $\nabla F(x,\lambda)$ of course, denotes differentiation with respect to the total variable (x,λ).

By a quasi-Newton method for approximating a stationary point x^* of $f:R^n \to R$ (i.e., $\nabla f(x^*) = 0$) we mean the iterative procedure

$$\bar{x} = x - B^{-1}\nabla f(x) \tag{1.1}$$

$$\bar{B} = \mathcal{B}(x,\bar{x},B). \tag{1.2}$$

The matrix $\mathcal{B}(x,\bar{x},B)$ is interpreted as an approximation to $\nabla^2 f(x^*)$. The three most popular quasi-Newton methods are

Newton's Method:

$$\mathcal{B}(x,\bar{x},B) = \nabla^2 f(\bar{x}) \ . \tag{1.3}$$

Discrete Newton's Method:

$$\beta(x,\bar{x},B) = \left(\frac{1}{h}\left[\frac{\partial f(\bar{x}+he_i)}{\partial x_i} - \frac{\partial f(\bar{x})}{\partial x_i}\right]\right) \tag{1.4}$$

where $e_1,\ldots,e_n$ are the natural basis vectors for R^n, h is a small positive scalar (ideally somewhere near the square root of the machine tolerance of the particular computer system being used) and (a_{ij}) denotes the matrix whose i,j-th component is a_{ij}.

Secant Methods:

$$\beta(x,\bar{x},B) = \beta_S(s,y,B) \tag{1.5}$$

where $s = \bar{x} - x$, $y = \nabla f(\bar{x}) - \nabla f(x)$ and β_S satisfies the secant equation

$$\beta_S(s,y,B)s = y \ . \tag{1.6}$$

At the present time the most popular secant update is the BFGS given by

$$\bar{B} = B + yy^T/y^Ts - Bss^TB/s^TBs \tag{1.7}$$

and in inverse form by

$$\bar{H} = H - [sy^TH + (Hy - s)s^T]/s^Ty + ss^T(y^THy)/(s^Ty)^2 \tag{1.8}$$

where $H = B^{-1}$ and $\bar{H} = \bar{B}^{-1}$. For more detail, see Dennis and Moré (1977).

In this paper we are interested in algorithms for solving the constrained optimization problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g(x) = 0 \end{aligned} \tag{1.9}$$

where $f:R^n \to R$ and $g:R^n \to R^m$ with $m \leq n$. In conjunction with problem (1.9)

we consider the

Lagrangian Function

$$\ell(x,\lambda) = f(x) + \lambda^T g(x) \qquad x \in R^n, \quad \lambda \in R^m \quad . \tag{1.10}$$

Let x^* be a local solution of problem (1.9) . It is classical that under the very mild regularity assumptions that f and g are continuously differentiable at x^* and $\nabla g_1(x^*),\ldots,\nabla g_m(x^*)$ are linearly independent there exists a unique Lagrange multiplier λ^* such that (x^*,λ^*) is a solution of the nonlinear system

$$\nabla \ell(x,\lambda) = 0 \quad . \tag{1.11}$$

Observe that

$$\nabla \ell(x,\lambda) \equiv \begin{pmatrix} \nabla_x \ell(x,\lambda) \\ \nabla_\lambda \ell(x,\lambda) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + \nabla g(x)\lambda \\ g(x) \end{pmatrix} \tag{1.12}$$

and

$$\nabla^2 \ell(x,\lambda) = \begin{pmatrix} \nabla_x^2 \ell(x,\lambda) & \nabla g(x) \\ \nabla g(x)^T & 0 \end{pmatrix} . \tag{1.13}$$

By the extended problem corresponding to problem (1.9) we mean problem (1.11). In other words, by the extended problem we mean the problem of finding a stationary point of the Lagrangian functional. Moreover, it is also classical that (x^*,λ^*) corresponds to a saddle point of the extended problem. Indeed, under the assumption of regularity we have that $\nabla g(x^*) \neq 0$ and it is not difficult to demonstrate that this implies that $\nabla^2 \ell(x^*,\lambda^*)$ is necessarily indefinite and the saddle point behavior follows. The motivation for our use of the terminology 'extended' should be clear from the fact that the dimension of problem (1.11) (i.e., number of independent variables) is $n+m$.

The extended problem will play a fundamental role in our development and actually has been in the background of the derivation of many algorithms whether the researcher was aware of it or not. Our basic assumptions for the analysis given in this paper will be the standard Newton's method assumptions for the extended problem. Specifically, we assume

(1.14) (i) f and g have three continuous derivatives at x^*

and

(1.15) (ii) $\nabla^2 \ell(x^*,\lambda^*)$ is invertible.

The latter assumption (ii) is often referred to as nonsingularity of the solution x^* and clearly implies the regularity of x^*.

Before we leave this introductory section we would like to make some observations concerning the historical development of algorithms for constrained minimization problems. There has been an excessive amount of effort spent on attacking the constrained minimization problem by solving a sequence of unconstrained minimization problems. These approaches are usually rationalized by arguing that in this way one is able to utilize the excellent new quasi-Newton algorithms for unconstrained minimization. Specifically, we first witnessed considerable activity in penalty function methods and then substantial activity in multiplier methods. This philosophy is wrong and has retarded the progress of constrained optimization theory. Fortunately, we have finally arrived at the point where workers in the area of constrained optimization are no longer wearing the straightjacket of sequential unconstrained minimization formulations.

The entire course of events, including the recent activity in the area of quasi-Newton methods for constrained minimization, can be explained best in terms of the extended problem. To begin with, quasi-Newton methods are, in their purest form, algorithms for solving systems of nonlinear equations.

This means that with respect to nonlinear functionals they are algorithms for approximating stationary points and not necessarily just minimizers or maximizers. Indeed, this is the way they were presented earlier in this section. However, many researchers seem to be secure only when they are applying these algorithms to a minimization (or maximization) problem. Some of this security is understandable since in this latter case one can obtain step length control from a line search routine. Namely, (1.1) is replaced with

$$\bar{x} = x - \alpha B^{-1} \nabla f(x) \tag{1.16}$$

where α is an approximation to the solution of the one-dimensional problem

$$\underset{\alpha}{\text{minimize}}\ f(x - \alpha B^{-1} \nabla f(x)) \tag{1.17}$$

and f is the nonlinear functional for which a minimizer is sought.

There is no doubt that some form of step length control is needed for the effective implementation of quasi-Newton methods. However, this in no way implies that only extremum problems can be handled effectively. This brings us to our main point. For some reason the vast majority of research activity in constrained optimization has historically ignored the extended problem. Partial motivation for this is obviously the fact that the desired solution is known to be a saddle point. So, instead of solving one problem we have been led to solving an infinite sequence of problems. We were asked to accept this task simply because each problem in this sequence was a true minimization problem. Moreover, this acceptance represents a ten-year detour in the development of effective constrained optimization algorithms.

The truly fascinating aspect of this research area is that (without being aware of it) researchers have very recently suggested quasi-Newton

methods for constrained optimization problems which can be shown to be equivalent to a quasi-Newton method applied to the extended problem. These approaches will be discussed in detail in Section 5 . In summary, it is interesting that the circle has been completed and we are now at a point in the development of the theory that we would have been ten years ago, had it not been for the sequential unconstrained minimization detour.

2. The Fundamental Role of the Penalty Constant.

We will now show that the fundamental role of the penalty constant is one of transforming a nonconvex functional into a locally convex functional so that minimizers exist. In the penalty function method and the multiplier method we will be concerned with the sequential minimization of the following two functionals.

Penalty Function

$$P(x) = f(x) + \tfrac{1}{2}Cg(x)^T g(x) \qquad (C>0) \tag{2.1}$$

and

Augmented Lagrangian Function

$$L(x,\lambda) = f(x) + \lambda^T g(x) + \tfrac{1}{2}Cg(x)^T g(x) \qquad (C\geq 0) \tag{2.2}$$

where f and g are given in problem (1.9) and $\lambda \in R^m$. The key question here is whether these functionals have minimizers for a fixed λ and a fixed C. Straightforward calculations give

$$\nabla_x^2 \ell(x,\lambda) = \nabla^2 f(x) + \lambda_1 \nabla^2 g_1(x) + \ldots + \lambda_m \nabla^2 g_m(x) \tag{2.3}$$

where the Lagrangian ℓ is given by (1.10),

$$\nabla^2 P(x) = \nabla_x^2 \ell(x, Cg(x)) + C\nabla g(x)\nabla g(x)^T , \tag{2.4}$$

and

$$\nabla_x^2 L(x,\lambda) = \nabla_x^2 \ell(x, \lambda + Cg(x)) + C\nabla g(x)\nabla g(x)^T . \tag{2.5}$$

Let x^* be a local solution of problem (1.9) and λ^* its associated multiplier. It is known that $\nabla_x^2 \ell(x^*,\lambda^*)$ may be indefinite; hence the Lagrangian functional $\ell(x,\lambda^*)$ need not have a minimizer in x. However, the following theorem shows that for C sufficiently large, the Hessian of the augmented Lagrangian functional at (x^*,λ^*) is positive definite.

This means that the penalty functional will have a local minimizer in x provided $Cg(x)$ is near λ^*, while the augmented Lagrangian will have a minimizer in x provided $\lambda + Cg(x)$ is near λ^*.

Theorem 2.1. Let x^* be a local solution of problem (1.9) and let λ^* be its associated multiplier. Suppose conditions (1.14) and (1.15) hold. Then there exists $C^* > 0$ such that for all $C \geq C^*$ the matrix

$$\nabla_x^2 \ell(x^*, \lambda^*) + C \nabla g(x^*) \nabla g(x^*)^T \tag{2.6}$$

is positive definite.

Proof. We give a short proof due to Buys (1972). Let A denote $\nabla_x^2 \ell(x^*, \lambda^*)$ and G denote $\nabla g(x^*)$. Consider the compact set

$$S = \{\eta \in R^n : \|\eta\| = 1\} . \tag{2.7}$$

If the theorem is not true, then there exists a sequence $\{\eta_k\}$ in S such that

$$\eta_k^T (A + kGG^T) \eta_k \leq \frac{1}{k} , \qquad k = 1, 2, \ldots . \tag{2.8}$$

But $\{\eta_k\}$ has a convergent subsequence converging to $\bar{\eta} \in S$. Since $\eta^T G G^T \eta \geq 0 \ \forall \ \eta$ it follows that

$$\bar{\eta}^T G G^T \bar{\eta} = 0 \quad \text{and} \quad \bar{\eta}^T A \bar{\eta} \leq 0 . \tag{2.9}$$

The first part of (2.9) implies that $G^T \bar{\eta} = 0$. The standard second order necessary conditions (see Chapter 2 of Fiacco and McCormick (1968)) imply that $\bar{\eta}^T A \bar{\eta} \geq 0$, which together with the second part of (2.9) implies that

$$\bar{\eta}^T A \bar{\eta} = 0 . \tag{2.10}$$

It follows that the problem

(2.11) $$\text{minimize } \{\tfrac{1}{2}\eta^T A\eta : G^T\eta = 0\}$$

has a solution at $\bar{\eta}$. Hence there exists a multiplier $\mu \in R^m$ such that

$$A\bar{\eta} + G\mu = 0, \quad G^T\bar{\eta} = 0 .$$

This implies that the columns of $\nabla^2 \ell(x^*,\lambda^*)$ (see (1.13)) are linearly independent and contradicts assumption (1.15). ∎

The main problems with this theorem are that it does not tell us what C^* is and it only gives us convexity near the solution.

3. The Penalty Function Method.

The penalty function method for approximating x^*, the solution of problem (1.9), is essentially the iterative procedure

(3.1) calculate $\bar{x} = \arg\min_x P(x)$

(3.2) choose $\bar{C} > C$

where, of course, $P(x)$ is the penalty function given by (2.1) The penalty function method has been known for years and among the first to consider it was Courant (1943). The approach was developed and publicized to a large extent by Fiacco and McCormick (1968), although numerous other authors have contributed to the subject.

From (3.1) we must have

$$\nabla P(\bar{x}) = \nabla f(\bar{x}) + \nabla g(\bar{x}) C g(\bar{x}) = 0 \quad . \tag{3.3}$$

Considering the first order necessary conditions (1.11) we are led to the conclusion that $Cg(\bar{x})$ is an approximation to the Lagrange multiplier λ^* associated with x^* . Moreover, if $x \to x^*$, then $Cg(x) \to \lambda^*$; but $g(x) \to 0$ so necessarily $C \to +\infty$ and we have convergence only if the penalty constant becomes infinite. The following main convergence theorem for the penalty function method is due to Polyak (1971).

Theorem 3.1. There exists a constant $\hat{C}$ such that for every $C > \hat{C}$ the penalty function $P(x)$ has a locally unique minimizer, say $x(C)$. Furthermore, there exists a constant $M > 0$ such that

$$\|x(C) - x^*\| \leq M/C \qquad \forall\ C > \hat{C} \tag{3.4}$$

and

$$\|Cg(x(C)) - \lambda^*\| \leq M/C \quad \forall\ C > \hat{C} \ . \tag{3.5}$$

Proof. For a proof see Polak (1971) or Bertsekas (1976). Bertsekas' proof is slightly more general and his conditions are implied by our assumptions (1.14) and (1.15).

Corollary 3.1. Suppose that the initial penalty constant in the penalty function method is larger than $\hat{C}$ in Theorem 3.1. Then the penalty function method is convergent if and only if $C \to +\infty$.

Corollary 3.2. As the penalty function method converges the numerical conditioning (as measured by the condition number of the Hessian of $P(x)$) becomes arbitrarily bad.

Proof. As $C \to \infty$ any norm of the matrix in (2.6) will become infinite. ∎

It is very important to observe that the penalty function method is not really an iterative procedure. Namely, $\bar{x}$ does not depend on x unless the choice of $\bar{C}$ depends on x . The penalty constant C actually plays a role analogous to the role the mesh spacing plays in the solution of differential and integral equations by finite differences. Specifically, we can get arbitrarily good accuracy by choosing the initial penalty constant sufficiently large. The question that should be asked is: Why minimize $P(x)$ for various values of C ? Obviously, we need only minimize $P(x)$ for the largest value of C that we are interested in. Of course, the numerical conditioning of the problem enters in (as it does in finite differences) and it is not clear what the optimal value of C should be. Our point here is that the nature of the penalty function method is significantly different than that of a standard iterative procedure and is similar, from a philosophical point of view, to the discretization methods in differential equations. This point seems not to have been appreciated in the literature, yet its implications are incredible as we shall now see.

To begin with, the question of local convergence is meaningless. Indeed, the iterates are independent of the initial iterate and as long as the initial penalty constant is large enough so that the minimization problems are well defined we will always obtain convergence. Is it fair to say that the algorithm is globally convergent? What about convergence rates? Well, since each iterate depends only on the penalty constant and we are free to choose the penalty constants, it is clear that from a theoretical point of view we can obtain a convergence rate of any order. These statements are surprising but mathematically true. However, the rub is Corollary 3.2, i.e., round-off error, and these elaborate claims would be impossible to demonstrate on any finite-precision computer. In summary, we conclude that the penalty constant has an extremely important and unique role in the penalty function method.

4. The Multiplier Method.

The multiplier method was originally proposed by Hestenes (1969) and independently in different but equivalent forms by Powell (1969) and Haarhoff and Buys (1970). The rationale for the multiplier method is to give a method which is as effective as the penalty function method but does not suffer from the numerical ill-conditioning of the penalty function method.

The multiplier method for problem (1.9) as suggested by Hestenes, consists of the iterative procedure: Given an initial λ and $C>0$

(4.1) calculate $\bar{x} = \arg\min_x L(x,\lambda)$,

(4.2) let $\bar{\lambda} = \lambda + Cg(\bar{x})$,

(4.3) choose $\bar{C} > 0$,

where the augmented Lagrangian $L(x,\lambda)$ is given by (2.2). Haarhoff and Buys suggested the same algorithm except that instead of the multiplier update formula (4.2) they suggested the multiplier update formula

(4.4) $$\bar{\lambda} = -[\nabla g(\bar{x})^T \nabla g(\bar{x})]^{-1} \nabla g(\bar{x})^T \nabla f(\bar{x}) \ .$$

Proposition 4.1. The multiplier methods with the multiplier update formula (4.2) and the multiplier method with the multiplier update formula (4.4) give identical (x,λ) iterates.

Proof. The proof is straightforward.

The multiplier method motivated some very beautiful duality theory for nonlinear programming problems. This theory was developed independently by Buys (1972), Luenberger (1973) and Rockafellar (1973). We shall present Buys' approach below. Motivated by his duality theory, Buys suggested that

the multiplier update formula (4.2) be replaced by the multiplier update formula

$$\bar{\lambda} = \lambda + [\nabla g(\bar{x})^T \nabla_x^2 L(\bar{x},\lambda)^{-1} \nabla g(\bar{x})]^{-1} g(\bar{x}) \quad . \tag{4.5}$$

Tapia (1977) proposed the general multiplier update formula

$$\bar{\lambda} = \lambda + [\nabla g(\bar{x})^T D \nabla g(\bar{x}) + A]^{-1} [g(\bar{x}) - \nabla g(\bar{x})^T D \nabla_x L(\bar{x},\lambda)] \tag{4.6}$$

where A and D are $m \times m$ and $n \times n$ matrices which may depend on x, λ and C. He showed that all previously suggested multiplier update formulas were special cases of this general formula. For more detail on multiplier update formulas see Sections 2 and 4 of Tapia (1977).

We first analyze the role of the multiplier update formula (4.2) and the specific role of the penalty constant in this formula. This will be accomplished by looking at Buys' nonlinear duality theory. Let x^* be the nonsingular solution of problem (1.9) with associated Lagrange multiplier λ^*. Assume that $C > C^*$ where C^* is given in Theorem 2.1. By the implicit function theorem (p. 128 of Ortega and Rheinboldt (1970)), there exists a neighborhood W of λ^* and a function $x: W \subset R^m \to R^n$ with the following properties:

$$x(\lambda^*) = x^*, \tag{4.7}$$

$$\nabla_x L(x(\lambda),\lambda) = 0, \tag{4.8}$$

$$\nabla x(\lambda) = -\nabla g(x(\lambda))^T \nabla_x^2 L(x(\lambda),\lambda)^{-1}, \tag{4.9}$$

and

$$h(\lambda) = \min_x L(x,\lambda) \tag{4.10}$$

is well defined on W. Problem (1.9) is called the primal problem. The dual problem is defined below:

(4.11) $$\max_{\lambda} h(\lambda) .$$

In (4.11) we have tacitly assumed that λ is restricted to the open set W. Since

(4.12) $$h(\lambda) = L(x(\lambda),\lambda)$$

using (4.8) and (4.9) we see that

(4.13) $$\nabla h(\lambda) = g(x(\lambda)),$$

(4.14) $$\nabla^2 h(\lambda) = -\nabla g(x(\lambda))^T \nabla_x^2 L(x(\lambda),\lambda)^{-1} \nabla g(x(\lambda)).$$

From Theorem 2.1, $\nabla_x^2 L(x^*,\lambda^*)$ is positive definite; so $\nabla^2 h(\lambda^*)$ is negative definite. Combining these remarks leads us to the following duality principle.

Theorem 4.2. (Local Duality). If x^* solves the primal problem, then its associated Lagrange multiplier λ^* solves the dual problem and x^* can be obtained from λ^* as the solution of $\min_x L(x,\lambda^*)$.

Theorem 4.3. The multiplier method with multiplier update formula (4.2) is the gradient method with steplength C applied to the dual problem and the multiplier method with multiplier update formula (4.5) is Newton's method applied to the dual problem.

Let us consider the multiplier method as given by (4.1) - (4.3). One function that the penalty constant has is to make step (4.1) well defined as dictated by Theorem 2.1. Another role that it plays, according to (4.2) and Theorem 4.3, is that of acting as the step length in the gradient method applied to the dual problem. This latter role tells us from gradient method theory we will be able to obtain local linear convergence for a range of penalty constants. This range will depend on the eigenvalue structure

of the Hessian matrix $\nabla_x^2 L(x^*,\lambda^*)$. Moreover, we are led to believe that, in contrast to the penalty function method, we cannot let the penalty constant grow arbitrarily fast. In fact, at this point we do not even know if it is possible to let the penalty constant become infinite and if anything would be gained by such a choice.

We now consider local convergence and convergence rates for the multiplier method. Bertsekas (1976) generalized Polyak's theorem (Theorem 3.1) to include the multiplier method in the following manner. As before, we are assuming conditions (1.14) and (1.15) and x^* is a local solution of problem (1.9) with associated multiplier λ^*.

Theorem 4.4. Let S be a bounded subset of R^m which contains λ^* as an interior point. Then there exists a constant $\hat{C}$ such that for $C>\hat{C}$ and $\lambda \in S$ the augmented Lagrangian $L(x,\lambda)$ has a locally unique minimizer, say $x(\lambda,C)$. Furthermore, there exists a constant $M>0$ such that

$$\|x^* - x(\lambda,C)\| \leq \frac{M}{C}\|\lambda - \lambda^*\| \tag{4.15}$$

and

$$\|\lambda^* - \bar{\lambda}(\lambda,C)\| \leq \frac{M}{C}\|\lambda - \lambda^*\|, \quad \forall\ C>\hat{C} \text{ and } \forall\ \lambda \in S \tag{4.16}$$

where

$$\bar{\lambda}(\lambda,C) = \lambda + Cg(x(\lambda,C)) \ . \tag{4.17}$$

Several questions immediately come to mind concerning local convergence and convergence rates of the multiplier method and we shall attempt to answer these questions in the remainder of this section. To begin with, from (4.16) of Theorem 4.4, we expect to be able to analyze convergence of λ in terms of Q-convergence. However, (4.15) does not lead to the same conjecture in terms of the convergence of x. In fact, on the surface it looks as if one might have to settle for an analysis in terms of R-convergence. For definitions

of these convergence notions see Section 8 of Tapia (1977) and for more detail see Chapter 9 of Ortega and Rheinboldt (1970). The following result proved in Section 9 of Tapia (1977) gives us the satisfaction that the convergence in x and λ is essentially the same.

Proposition 4.2. Suppose that the multiplier method with an arbitrary Lagrange multiplier update formula and an arbitrary bounded sequence of penalty constants $\{C^k\}$ such that $C^k \geq \hat{C}$ generates the sequences $\{x^k\}$ and $\{\lambda^k\}$. Then $\lambda^k \to \lambda^*$ with Q-order q if and only if $x^k \to x^*$ with Q-order q.

Proof. The proof of this result is given in Section 9 of Tapia (1977).

As a direct consequence of Theorem 4.3 and Proposition 4.2 we have the following convergence result for the multiplier method.

Proposition 4.3. For any given initial estimate of the Lagrange multiplier λ there exists a penalty constant $\hat{C} > 0$ such that the multiplier method with fixed penalty constant $C > \hat{C}$ is Q-linearly convergent in x and in λ.

Observe that in the multiplier method the penalty constant cannot be increased arbitrarily fast as it can in the penalty function method. If it grows too fast, then $\bar{\lambda}(\lambda, C)$ given by (4.17) will become excessively large (i.e., it will not remain in the set S in Theorem 4.3) and the convergence will suffer. It is clear that the increase in C must be balanced with the decrease in $g(x)$. However, from (4.16) we see that Q-superlinear convergence would result if it were possible to let the sequence of penalty constants become unbounded. This latter consideration is the subject of the following proposition.

Proposition 4.4. It is possible to choose $\{C^k\}$ so that $C^k \to \infty$ and the

multiplier method with penalty constant $\{c^k\}$ is convergent in x and λ.

Proof. The proof follows directly from Theorem 4.3. Specifically, let $S = \{\lambda: \|\lambda - \lambda^*\| \leq 1\}$, and choose $c^0 > \hat{c}$ so that $M/c^0 < \frac{1}{2}$ and choose λ^0 so that $\|\lambda^* - \lambda^0\| \leq \frac{1}{2}$. Then

$$\|x^* - x^k\| \leq \frac{1}{2^k} \quad \text{and} \quad \|\lambda^* - \lambda^k\| \leq \frac{1}{2^k} \tag{4.18}$$

as long as $c^k \geq c^0$ and

$$c^k \|g(x^k)\| \leq \tfrac{1}{2}. \tag{4.19}$$

From (4.19)it is clear that we can choose $\{c^k\}$ so that $c^k \to \infty$. ∎

We are concerned with the role of the penalty constant in the multiplier method. So far we have seen that it allows one to obtain Q-linear convergence and Q-superlinear convergence if it becomes infinite. Recall that in the penalty function method we obtained convergence if and only if the penalty constant became infinite. The situation would be mathematically satisfying if the analogous situation for the multiplier method was such that we were able to obtain superlinear convergence if and only if the penalty constant became infinite. The following proposition establishes this fact. For the purposes of this result we will assume that $\hat{c} = 0$.

Proposition 4.5. Suppose that the multiplier method with penalty constants c^k is convergent. Then the convergence is Q-superlinear in λ if and only if $c^k \to \infty$.

Proof. The 'if' part follows directly from Theorem 4.3. The 'only if' part was first demonstrated in Section 9 of Tapia (1977). The proof is sufficiently interesting and novel that we will reproduce it here. Assume that λ^k converges Q-superlinearly to λ^*. We are concerned with the iteration

(4.20) $$\lambda^{k+1} = S(\lambda^k, C^k)$$

where

(4.21) $$S(\lambda, C) = \lambda + Cg(x, (\lambda))$$

and $x(\lambda)$ is as in (4.7) - (4.14) . Now for a fixed C we see from (4.9) that

(4.22) $$S'_\lambda(\lambda^*, C) = I - C\nabla g(x^*)^T \nabla^2_x L(x^*, \lambda^*)^{-1} \nabla g(x^*) .$$

Let $A = \nabla^2_x \ell(x^* \lambda^*)$ and $G = \nabla g(x^*)$ so that

(4.23) $$\nabla^2_x L(x^*, \lambda^*) = A + CGG^T$$

and from (4.22)

(4.24) $$S'_\lambda(\lambda^*, C) = I - CG^T(A + CGG^T)^{-1} G .$$

From the Sherman-Morrison-Woodbury formula (page 50 of Ortega and Rheinboldt (1970)) we obtain

(4.25) $$(A + CGG^T)^{-1} = A^{-1} + CA^{-1} G(I + CG^T A^{-1} G)^{-1} G^T A^{-1}$$

so that

(4.26) $$S'_\lambda(\lambda^*, C) = [I + C\nabla g(x^*)^T \nabla^2_x \ell(x^*, \lambda^*)^{-1} \nabla g(x^*)]^{-1} .$$

Observe that for $C \geq 0$ the matrix $S'_\lambda(\lambda^*, C)$ is positive definite and hence invertible.

From McLeod's mean-value theorem (see Tapia (1971)) we have

(4.27) $$\lambda^{k+1} - \lambda^* = S(\lambda^k, C^k) - \lambda^* = \sum_{i=1}^{m} t_i \, S'_\lambda(\lambda^* + \theta_i(\lambda^k - \lambda^*), C^k)(\lambda^k - \lambda^*)$$

where

$$0 < \theta_i < 1, \quad t_i \geq 0, \quad \sum_{i=1}^{m} t_i = 1 .$$

Suppose that a subsequence of $\{C^k\}$ (also denoted by $\{C^k\}$) converges to $K < +\infty$. Let

(4.28) $$s_k = (\lambda^k - \lambda^*)\|\lambda^k - \lambda^*\|^{-1} .$$

By compactness, $\{s_k\}$ has a subsequence (which we also denote by $\{s_k\}$) which converges to $s^* \neq 0$. Dividing both sides of (4.20) by $\|\lambda^k - \lambda^*\|$, recalling the definition of Q-superlinear convergence, and letting $k \to \infty$, we obtain

(4.29) $$S'_\lambda(\lambda^*, K)(s^*) = 0 .$$

However, this is a contradiction, since $S'_\lambda(\lambda^*, K)$ is invertible. It follows that $c^k \to \infty$ and this proves the proposition. ∎

Our analysis of the role of the penalty function in the multiplier method is now complete.

5. The Quasi-Newton Methods for Constrained Optimization.

Let us summarize our presentation up to this point. We have observed that in the penalty function method the price one pays for convergence is a deterioration in numerical conditioning, since the penalty constant must go to infinity. In the multiplier method, the price one pays for superlinear convergence is also a deterioration in numerical conditioning, since again the penalty function must go to infinity. Clearly the stage is set for an algorithm which will give superlinear convergence without a corresponding deterioration in numerical conditioning. Such algorithms exist and will now be presented.

There are essentially three philosophies for extending quasi-Newton methods from unconstrained optimization to constrained optimization. These philosophies consist of the multiplier extension quasi-Newton methods, the multiplier update quasi-Newton methods and the quadratic programming quasi-Newton methods. In Tapia (1978) these three approaches are shown to be equivalent for problem (1.9). Our presentation will follow Tapia (1978) closely.

For the purposes of this section, by the extended problem we will mean the problem of finding a stationary point of the augmented Lagrangian given by (2.2), i.e., the problem of finding (x,λ) such that

$$\nabla L(x,\lambda) = 0 \ . \tag{5.1}$$

As in previous sections we denote a local solution of problem (1.9) by x^* and its associated Lagrange multiplier by λ^* .

Multiplier Extension Quasi-Newton Methods

By a multiplier extension quasi-Newton method for problem (1.9) we mean the iterative procedure

$$\begin{pmatrix} \bar{x} \\ \bar{\lambda} \end{pmatrix} = \begin{pmatrix} x \\ \lambda \end{pmatrix} - B^{-1}\nabla L(x,\lambda) \tag{5.2}$$

$$\bar{B} = \begin{pmatrix} \bar{B}_x & \nabla g(\bar{x}) \\ \nabla g(\bar{x})^T & 0 \end{pmatrix} \tag{5.3}$$

where $\bar{B}_x$ is an approximation to $\nabla_x^2 L(x^*,\lambda^*)$.

The multiplier <u>secant methods</u> result by choosing

$$\bar{B}_x = \mathcal{B}_S(s,y,B_x) \tag{5.4}$$

where $s = \bar{x} - x$, $y = \nabla_x L(\bar{x},\bar{\lambda}) - \nabla_x L(x,\bar{\lambda})$, B_x is the current approximation to $\nabla_x^2 L(x^*,\lambda^*)$ and $\mathcal{B}_S$ is one of the popular secant updates, e.g., BFGS . The multiplier extension secant method played an important role in the theory developed in Tapia (1977) and (at present) we have no references to earlier usage.

Observe that a straightforward quasi-Newton method would consist of approximating the entire Hessian matrix

$$\nabla^2 L(x,\lambda) = \begin{pmatrix} \nabla_x^2 L(x,\lambda) & \nabla g(x) \\ \nabla g(x)^T & 0 \end{pmatrix} . \tag{5.5}$$

Our multiplier extension quasi-Newton is not that naïve. Specifically, we have taken advantage of a certain amount of structure that the problem has to offer by only approximating the component of $\nabla^2 L(x,\lambda)$ in (5.5) which contains second order information. Basically, it seems inefficient to approximate first order information that has already been calculated exactly, or even worse yet, to approximate the zero component in $\nabla^2 L(x,\lambda)$. Carrying this line of reasoning one step further we observe that

$$\nabla_x^2 L(x^*,\lambda^*) = \nabla_x^2 \ell(x^*,\lambda^*) + C\nabla g(x^*)\nabla g(x^*)^T . \tag{5.6}$$

Consequently, although we have taken advantage of some structure we have more, i.e., we need not approximate the first order information in (5.6). Tapia (1978) referred to this additional structure as superstructure (see Section 7 of Tapia (1978)). We modify the multiplier extension secant method to take advantage of superstructure by replacing $\bar{B}_x$ in (5.4) by

$$\bar{B}_x = M_S(s,y,M) + C\nabla g(\bar{x})\nabla g(\bar{x})^T \tag{5.7}$$

where

$$s = \bar{x} - x \ , \tag{5.8}$$

$$y = \nabla_x \ell(\bar{x},\bar{\lambda}) - \nabla_x \ell(x,\bar{\lambda}), \tag{5.9}$$

M is the current approximation to $\nabla_x^2 \ell(x^*,\lambda^*)$ and M_S is a secant update.

Multiplier Update Quasi-Newton Methods

By a multiplier update quasi-Newton method for problem (1.9) we mean the iterative procedure

$$\bar{\lambda} = (\nabla g^T B^{-1} \nabla g)^{-1}(g - \nabla g^T B^{-1} \nabla f) - Cg \tag{5.10}$$

$$\bar{x} = x - B^{-1}\nabla_x L(x,\bar{\lambda}) \tag{5.11}$$

$$\bar{B} = \mathcal{B}(x,\bar{x},\lambda,\bar{\lambda},B) \tag{5.12}$$

where $\mathcal{B}(x,\bar{x},\lambda,\bar{\lambda},B)$ is an approximation to $\nabla_x^2 L(x^*,\lambda^*)$. The multiplier update secant methods result by choosing

$$\mathcal{B}(x,\bar{x},\lambda,\bar{\lambda},B) = B_S(s,y,B) \tag{5.13}$$

where $s = \bar{x} - x$, $y = \nabla_x L(\bar{x},\bar{\lambda}) - \nabla_x L(x,\bar{\lambda})$ and B_S is a secant update .

We take advantage of superstructure by replacing (5.13) with (5.7) - (5.9). The multiplier update secant methods were proposed by the author in Tapia (1977) and in that paper inequality constraints were handled via a

slack variable. Independently, Han (1977) proposed secant methods for problems with equality and inequality constraints which use an intermediate quadratic program to solve for the multipliers. In the case of problem (1.9) (no inequality constraints) it is a simple matter to show that Han's quadratic program reduces to (5.10) and hence his algorithm reduces to the multiplier update secant method. Glad (1976), also independently, proposed the multiplier update secant method. He used an active constraint philosophy to handle inequality constraints. All three papers established superlinear convergence.

Quadratic Programming Quasi-Newton Methods

By a quadratic programming quasi-Newton method for problem (1.9) we mean the iterative procedure

$$\bar{x} = x + \Delta x \tag{5.14}$$

$$\bar{B} = \mathcal{B}(x, \bar{x}, B) \tag{5.15}$$

where $\mathcal{B}(x,\bar{x},B)$ is an approximation to $\nabla_x^2 L(x^*,\lambda^*)$ and Δx is a solution of the quadratic program

$$\min_{\Delta x} q(\Delta x) = F(x) + \nabla F(x)^T \Delta x + \tfrac{1}{2}\Delta x^T B \Delta x \tag{5.16}$$

$$\text{subject to } \nabla g(x)^T \Delta x + g(x) = 0 .$$

with

$$F(x) = f(x) + \frac{c}{2} g(x)^T g(x) . \tag{5.17}$$

The quadratic programming secant methods result by choosing

$$\mathcal{B}(x,\bar{x},B) = \mathcal{B}_S(s,y,B) \tag{5.18}$$

where $s = \bar{x} - x$, $y = \nabla_x L(\bar{x},\lambda_{QP}) - \nabla_x L(x,\lambda_{QP})$, $\mathcal{B}_S$ is a secant update and

λ_{QP} is the multiplier obtained in the solution of the quadratic program (5.16) .

This form of the quadratic programming quasi-Newton method was introduced by Garcia Palomares and Mangasarian (1976). Han (1976) added some analysis and specific secant updates and Powell (1977),(1978) added further refinements and analysis.

As before, we take advantage of superstructure by replacing (5.18) with (5.7) - (5.9) .

Theorem 5.1. The multiplier extension secant method, the multiplier update secant method, and the quadratic programming secant method are equivalent in the sense that they generate identical (x,λ) iterates. Moreover, for $C \geq \hat{C}$ where $\hat{C}$ is given by Theorem 2.1 these methods are locally Q-superlinearly convergent in the variable (x,λ) .

Proof. The proof of the local Q-superlinear convergence is somewhat involved and the reader interested in details is referred to Section 10 of Tapia (1977). The equivalence proof is instructive and short so we will reproduce it. The first line of (5.2) is exactly (5.11). By substituting (5.11) into the second line of (5.2) we arrive at

$$\nabla g(x)^T B_x^{-1} \nabla_x L(x,\bar{\lambda}) = g(x) \quad . \tag{5.19}$$

Solving (5.19) for $\bar{\lambda}$ leads directly to (5.10) and establishes the equivalence between the multiplier extension and the multiplier update secant methods. Now, problem (5.16) is equivalent to

$$\nabla f(x) + B\Delta x + \nabla g(x)[\lambda_{QP} + Cg(x)] = 0 \tag{5.20}$$

$$\nabla g(x)^T \Delta x + g(x) = 0 \quad . \tag{5.21}$$

From (5.20) we see that

$$\Delta x = -B^{-1}\nabla_x L(x,\lambda_{QP}) \quad . \tag{5.22}$$

Substituting (5.22) into (5.21) and solving for λ_{QP} gives

$$\lambda_{QP} = \bar{\lambda} \tag{5.23}$$

where $\bar{\lambda}$ is given by (5.10) .

It certainly seems reasonable to question the choice of working with the penalty function F given by (5.17) in the constant and first order terms of (5.16) and using the augmented Lagrangian in the second order term, in contrast to working with the augmented Lagrangian in all three terms. Such an approach would require the explicit use of an approximate multiplier. This leads to the following modification of the quadratic programming secant method

$$\bar{x} = x + \Delta x \tag{5.24}$$

$$\bar{\lambda} = \lambda + \Delta\lambda \tag{5.25}$$

$$\bar{B} = \mathcal{B}_S(s,y,B) \tag{5.26}$$

where Δx and $\Delta\lambda$ are the solution and corresponding multiplier obtained from solving the quadratic program

$$\min_{\Delta x} \hat{q}(\Delta x) = L(x,\lambda) + \nabla_x L(x,\lambda)^T \Delta x + \tfrac{1}{2}\Delta x^T B \Delta x \tag{5.27}$$

$$\text{subject to } \nabla g(x)^T \Delta x + g(x) = 0$$

with $s = \bar{x} - x$, $y = \nabla_x L(\bar{x},\bar{\lambda}) - \nabla_x L(x,\bar{\lambda})$ and as before $\mathcal{B}_S$ is a secant update.

Proposition 5.1. The quadratic programming secant method (5.14) - (5.18) with λ_{QP} and the modified quadratic programming secant method (5.24) - (5.27) are equivalent. Specifically $\bar{\lambda}$ in (5.25) and λ_{QP} are the same for any value

of λ in (5.27); consequently these two algorithms generate identical (x,λ) iterates.

Proof. An argument similar to that used in the proof of Theorem 5.1 can be used to show that

$$\Delta\lambda = \lambda_{QP} - \lambda \tag{5.28}$$

and the result follows. ∎

Since the algorithm (5.24) - (5.27) is independent of the particular choice for λ one may as well choose $\lambda = 0$ and work with (5.14) - (5.18). The choice $C = 0$ also has special significance as we will now demonstrate.

The role of the penalty constant should now be clear. Namely, it allows one to obtain a positive definite Hessian $\nabla_x^2 L(x^*,\lambda^*)$. Moreover, it is standard procedure to implement the BFGS secant method so that the approximate Hessians are always positive definite. So everything fits together nicely in the sense that we are approximating a positive definite matrix by a sequence of positive definite matrices. Let us now look closely at the superstructured versions. To begin with they are also equivalent since the proof of Theorem 5.1 only used the fact that the B_x matrix in these three algorithms was the same. However, we are now approximating $\nabla_x^2 \ell(x^*,\lambda^*)$, which is not necessarily positive definite, by matrices M which (as secant updates) are positive definite. Since M^{-1} exists, it is natural to question the role of C in this case, i.e., $\bar{B}_x$ as given by (5.7) will be well-defined even in the extreme case when $C = 0$. The question becomes, what does positive C buy us? The following theorem provides us with a very satisfactory answer to this question.

Theorem 5.2. The superstructured versions of the multiplier extension,

multipler update and quadratic programming secant methods generate identical (x,λ) iterates which are independent of the penalty constant C.

Proof. We will work with the quadratic programming method.

Observe that in the case $B = M + C\nabla g \nabla g^T$ the system (5.20) - (5.21) reduces to the system

$$M\Delta x + \nabla f(x) + \nabla g(x)\lambda_{QP} = 0 \tag{5.29}$$

$$\nabla g(x)^T \Delta x + g(x) = 0 \tag{5.30}$$

We are assuming that the initial M matrix is independent of C; hence, Δx and λ_{QP} obtained from (5.29) - (5.30) will be independent of C, and from (5.9) we see that $\bar{M}$ will be independent of C. ∎

We can say with some confidence that taking advantage of available structure is worthwhile since it obviously leads to better approximate Hessians. However, acceptance of this statement implies that there is no need for the penalty constant and we have followed the role of the penalty constant to its logical conclusion.

In the literature we have seen several authors argue that the penalty constant should not be used because it is difficult to choose and its use merely makes the algorithm messy. Of course, we have had to accept this denial of the penalty constant in the context that it was made; namely, with little confidence. Recently Bertocchi, Cavalli and Spedicato (1979) performed numerical experiments and concluded that the choice $C = 0$ was probably optimal. This form of demonstration carries considerable validity. Moreover, it is very satisfying to couple Theorem 5.2 to their finding and conclude $C = 0$ is obviously the choice. However, are we really seeing the complete picture? The message that we have presented is the following: AS LONG AS LINEARIZED CONSTRAINTS ARE SATISFIED, WE DON'T NEED A POSITIVE

DEFINITE HESSIAN AND $C = 0$ IS OPTIMAL IN OUR QUASI-NEWTON METHODS. This, of course, merely leads us to the question: should linearized constraints always be satisfied? Most would agree with us that near the solution they should be satisfied. However, far from the solution linear approximations are often misleading. We feel that the question of whether or not an augmented Lagrangian can be used in an effective manner far from the solution is still an open question. Recently there has been some interesting work along this line by Biggs (1978) and by Boggs and Tolle (1977).

At the time of this writing, the local theory for $C = 0$ is still behind that for $C > 0$, in spite of the numerical experiments which favor the former locally. Specifically, a local convergence theorem for these quasi-Newton methods with $C = 0$ does not exist (see Theorem 5.1). Powell (1978) proved that if the x-iterates in the quadratic programming secant method with $C = 0$ converged, then they did so R-superlinearly. Although Powell expressed some disappointment that he was not able to establish Q-superlinear convergence in x, this has not been demonstrated even in the case of $C > 0$. Observe that Theorem 5.1 demonstrates Q-superlinear convergence of the pair (x,λ), which in turn implies R-superlinear convergence of x and of λ separately. We believe that without some slight modification of the procedure used for approximating the multiplier (as was done in Tapia (1977)) R-superlinear convergence is all one will be able to demonstrate for the x variable alone. It would be worthwhile investigating whether Powell's proof for $C = 0$ could be modified to give Q-superlinear convergence of the pair (x,λ).

REFERENCES

[1] Bertocchi, M., Cavalli, E. and Spedicato, E. (1979. Computational Performance of Diagonalized Multiplier Quasi-Newton Methods for Nonlinear Optimization with Equality Constraints, Manuscript.

[2] Bertsekas, D. P. (1976). On Penalty and Multiplier Methods for Constrained Minimization, SIAM J. on Control and Optimization, Vol. 14, pp. 216-235.

[3] Biggs, M. D. (1978). On the Convergence of Some Constrained Minimization Algorithms Based on Recursive Quadratic Programming, J. Inst. Math. Applications, Vol. 21, pp. 67-81.

[4] Boggs, P. T. and Tolle, J. W. (1977). Asymptotic Analysis of the Saddle Point of a Two-Parameter Multiplier Function, Technical Report No. 77-6, Operations Research and Systems Analysis, Univ. of North Carolina, Chapel Hill, North Carolina 27514.

[5] Buys, J. D. (1972). Dual Algorithms for Constrained Optimization, Ph.D. Thesis, Rijksuniversiteit de Leiden.

[6] Dennis, J. E. and More', J. J. (1977). Quasi-Newton Methods: Motivation and Theory, SIAM Review, Vol. 10, pp. 46-89.

[7] Fiacco, A.V. and McCormick, G. P. (1965). Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley, New York.

[8[Garcia Palomares, U. M. and Mangasarian, O. L. (1976). Superlinearly Convergent Quasi-Newton Algorithms for Nonlinearly Constrained Optimization Problems, Mathematical Programming, Vol. 11, pp. 1-13.

[9] Glad, T. (1976). Properties of Updating Methods for the Multipliers in Augmented Lagrangians, Ph.D. thesis, Univ. of Lund, Lund, Sweden.

[10] Haarhoff, P. C. and Buys, J. D. (1970. A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints, Computing Journal, Vol. 13, pp. 178-184.

[11] Han, S. P. (1976). Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, Mathematical Programming, Vol. 11, pp. 263-282.

[12] Han, S. P. (1977). Dual Variable Metric Algorithms for Constrained Optimization, SIAM J. on Control and Optimization, Vol. 15, pp. 546-565.

[13] Hestenes, M. R. (1969). Multiplier and Gradient Methods, J. Optimization Theory Appl., Vol. 4, pp. 303-320.

[14] Luenberger, D.G. (1973). Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass.

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

[16] Polyak, B. T. (1971). The Convergence Rate of the Penalty Function Method, Z. Vycisl. Mat. i Mat. Fiz., Vol. 11, pp. 3-11.

[17] Powell, M. J. D. (1969). A Method for Nonlinear Constraints in Minimization Problems, in Optimization, R. Fletcher, ed., Academic Press, New York, pp. 283-298.

[18] Powell, M. J. D. (1977). A Fast Algorithm for Nonlinearly Constrained Optimization Calculation. Report CAMTP 77/HA 2, Cambridge University, England.

[19] Powell, M. J. D (1978). The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations, in Nonlinear Programming 3, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, pp. 27-63.

[20] Rockafellar, R. T. (1973). A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization, Math. Programming, Vol. 5, pp. 354-373.

[21] Tapia, R. A. (1971). The Differentiation and Integration of Nonlinear Operators, in Nonlinear Functional Analysis and Applications, ed. L. B. Rall, Academic Press, New York.

[22] Tapia, R. A. (1974a). Newton's Method for Optimization Problems with Equality Constraints, SIAM J. on Numerical Analysis, Vol. 11, pp. 874-886.

[23] Tapia, R. A. (1974b). A Stable Approach to Newton's Method for General Mathematical Programming Problems in R^n, Journal of Optimization Theory and Applications, Vol. 14, pp. 453-476.

[24] Tapia, R. A. (1977). Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization, Journal of Optimization Theory and Applications, Vol. 22, pp. 135-194.

[25] Tapia, R. A. (1978). Quasi-Newton Methods for Equality Constrained Optimization: Equivalence of Existing Methods and a New Implementation, in Nonlinear Programming 3, O. Mangasarian, R. Meyer and S. Robinson, eds., Academic Press, New York, pp. 125-164.