Algorithms for Nonlinear Minimization with Equality and Inequality Constraints Based on Lagrange Multipliers

Glad, Torkel

1975

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

3092

ALGORITHMS FOR NONLINEAR MINIMIZATION WITH EQUALITY AND INEQUALITY CONSTRAINTS BASED ON LAGRANGE MULTIPLIERS

TORKEL GLAD

Report 7503 February 1975
Lund Institute of Technology
Department of Automatic Control

ALGORITHMS FOR NONLINEAR MINIMIZATION WITH EQUALITY AND INEQUALITY CONSTRAINTS BASED ON LAGRANGE MULTIPLIERS.

T. Glad

Abstract.

Constrained minimization methods based on different ways of updating the Lagrange multipliers are studied. It is shown that methods can be designed to converge rapidly (linearly or superlinearly) to the minimum. The methods are tested on a number of numerical problems.

This work has been supported by the Swedish Institute of Applied Mathematics.

TABLE OF CONTENTS

1. INTRODUCTION

This report deals with the nonlinear optimization problem which can be formulated as follows

Find the minimum of $f(x)$
under the constraints
$g_i(x) = 0 \quad i=1,..,p$
$g_i(x) \leq 0 \quad i=p+1,..,m$
where x is an n-vector.

The vector $(g_1(x),..,g_m(x))^T$ will be denoted $g(x)$.

In a previous report (Glad,(1973)), methods based on Lagrange multipliers for solving minimization problems under equality constraints, were studied. Among others a method based on the function $F(x,u) = f(x) + u^T g(x) + c\, g(x)^T g(x)/2$ was considered. The method consisted of a sequence of unconstrained minimizations of the function $F(x,u)$ followed by the updating of u, using the formula $u^{(k+1)}=u^{(k)} - G_{uu}^{-1}G_u$, where $G_{uu}= - g_x F_{xx}^{-1} g_x^T$ and $G_u= g$. Since the unconstrained minimization was performed with a Quasi-Newton method, an approximation of F_{xx} was available and the second derivatives of f and g were not calculated. Recently Fletcher (1973) has published a method using this approach also for inequality constraints. One possible drawback of this type of method is that a full minimization of $F(x,u)$ is done before u is updated. In Fletcher (1973) it is shown that usually most of the computational effort is spent on this first minimization and therefore improved ways of updating u do not reduce the computation time as much as one would expect, Several algorithms where u is updated more often than at the end of a minimization exist. Fletcher has published algorithms, see Fletcher (1970), Fletcher and Lill (1970), where u is taken as a function of x.

Miele et al. have several different schemes, see e.g. Miele et al. (1971), for updating u at every iteration or after a cycle of iterations. Tripati and Narendra (1972) have used the updating formula $u^{(k+1)} = u^{(k)} + \alpha cg$ at each iteration and found that the damping parameter α, which is less than 1, is usually needed to prevent divergence.

In this report some theoretical questions concerning the properties of the methods are investigated. In section 2 local convergence is studied, while global convergence problems are dealt with in section 3. Finally some numerical experiments with different algorithms are presented in sections 4 and 5.

2 DISCUSSION OF LOCAL CONVERGENCE PROPERTIES:

2.1 Properties of the optimization problem.

Here some basic facts concerning the optimization problem

minimize $f(x)$
under the constraints
$g_i(x) = 0 \quad i=1,..,p$
$g_i(x) \leq 0 \quad i=p+1,..,m$

will be given. It will be assumed that the problem has a solution which is called $\bar{x}$. Let $A = (i_k,..,i_j)$ be the set of indices such that $g_i(\bar{x}) = 0$, $i \in (p+1,..,m)$, (the active inequality constraints) and $I = (i_r,..,i_q)$ the set of indices such that $g_i(\bar{x}) < 0$, $i \in (p+1,..m)$, (the set of inactive constraints) . It will also be assumed that the standard second order sufficiency conditions for $\bar{x}$ to be a local minimum are satisfied:

(i) The functions f and g are twice continuously differentiable in some open set containing $\bar{x}$.

(ii) The gradient vectors $\nabla g_i(\bar{x})$, $i \in \{1,..,p\} \cup A$, belonging to the equality constraints and active inequality constraints, are linearly independent. It then follows that there exists a vector of Lagrange multipliers $\bar{u} = (\bar{u}_1,..,\bar{u}_m)^T$ such that the following conditions hold

$$f_x(\bar{x}) + \bar{u}^T g_x(\bar{x}) = 0$$

$$\bar{u}_i \geq 0 \qquad \bar{u}_i g_i(\bar{x}) = 0 \qquad i=p+1,...,m$$

(iii) The second derivative with respect to x of the Lagrangian, $L = f + u^T g$, satisfies

$$z^T L_{xx}(\bar{x},\bar{u})\, z > 0$$

for all vectors $z \neq 0$ such that $\nabla g_i^T(\bar{x}) z = 0 \quad i \in \{1,\ldots,p\} \cup A$

(iv) $\bar{u}_i > 0$ for all $i \in A$ (strict complementarity)

The Lagrangian does not usually have a minimum with respect to x at $(\bar{x},\bar{u})$, only a stationary point. Therefore the following function is used, Fletcher (1973), Bertsekas (1973).

$$F(x,u) = f(x) + \sum_{i=1}^{p} \{(c_i g_i(x)+u_i)^2 - u_i^2\}/2c_i +$$

$$\sum_{i=p+1}^{m} \{(c_i g_i(x)+u_i)_+^2 - u_i^2\}/2c_i$$

where $(c_i g_i + u_i)_+ = \begin{cases} c_i g_i + u_i & \text{if } c_i g_i + u_i \geq 0 \\ 0 & \text{if } c_i g_i + u_i < 0 \end{cases}$

It is shown in the references mentioned above that, if the parameters c_i are chosen large enough, and the conditions (i) - (iv) are satisfied, then $F(x,\bar{u})$ has a local minimum with respect to x at $\bar{x}$.

2.2 Local convergence when u is updated at each iteration.

If the algorithm to be discussed succeeds in generating sequences $x^{(k)}$ and $u^{(k)}$, that converge to $\bar{x}$ and $\bar{u}$, then there is some k_0 such that $(c_i g_i(x^{(k)}) + u_i^{(k)}) > 0$ for $k > k_0$ if $i \in A$, and $(c_i g_i(x^{(k)}) + u_i^{(k)}) < 0$ for $i \in I$ and $k > k_0$, (it is assumed that c_i is held constant for $k > k_0$). It is then possible to include the active inequality constraints among the equality constraints and disregard the inactive inequality constraints, as far as local convergence is concerned. In what follows it will therefore be assumed that the constraint is $g(x) = 0$. To simplify notation all the constants c_i will be assumed to have the same value. Then $F(x,u)$ can be written

$$F(x,u) = f(x) + u^T g(x) + \frac{c}{2} g(x)^T g(x)$$

An algorithm where minimization of $F(x,u)$ and updating of u are done alternatingly can be described by

$$\text{A1} \qquad \begin{aligned} u^{(k+1)} &= h(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} - a^{(k)} \cdot (B^{(k)})^{-1} F_x^T(x^{(k)}, u^{(k+1)}) \end{aligned}$$

Here $B^{(k)}$ is an approximation of the second derivative F_{xx}. The step length $a^{(k)}$ is determined by some procedure for approximate minimization of $F(x^{(k)} - a \cdot (B^{(k)})^{-1} F_x^T, u^{(k+1)})$ with respect to a. Since many linear minimization algorithms first try a=1, it is of interest to study the special case a=1. It will be assumed that $B^{(k)}$ is updated according to the Davidon-Fletcher-Powell formula and that h is continuously differentiable and satisfies the natural condition $h(\bar{x}) = \bar{u}$. The algorithm is then

$$\text{A2} \qquad \begin{aligned} u^{(k+1)} &= h(x^{(k)}) \\ x^{(k+1)} &= x^{(k)} - (B^{(k)})^{-1} F_x^T(x^{(k)}, u^{(k+1)}) \\ B^{(k+1)} &= B^{(k)} + \frac{(y^{(k)} - B^{(k)} s^{(k)})(y^{(k)})^T + y^{(k)} (y^{(k)} - B^{(k)} s^{(k)})^T}{(y^{(k)})^T s^{(k)}} \\ &\quad - \frac{(s^{(k)})^T (y^{(k)} - B^{(k)} s^{(k)}) y^{(k)} (y^{(k)})^T}{((s^{(k)})^T y^{(k)})^2} \end{aligned}$$

$$\text{where } y^{(k)} = F_x^T(x^{(k+1)}, u^{(k+1)}) - F_x^T(x^{(k)}, u^{(k+1)})$$
$$s^{(k)} = x^{(k+1)} - x^{(k)}$$

To study this algorithm the following lemma is used.

<u>Lemma 1</u> (Dennis and Moré (1974))
Let M be a nonsingular matrix such that $||My - M^{-1}s|| < b||M^{-1}s||$ for some b, $0 \leq b \leq 1/3$, and some vectors y and s with $s \neq 0$. Then $y^T s > 0$ and $\bar{B}$ can be defined by

$$\bar{B} = B + \frac{(y-Bs)y^T + y(y-Bs)^T}{y^T s} - \frac{s^T(y-Bs)yy^T}{(y^T s)^2}$$

where B is symmetric. Let $||\cdot||_M$ be the matrix norm defined by $||Q||_M = ||MQM||_F$ where $||P||_F$ is the Frobenius norm $||P||_F = \sqrt{\sum_{ij}|p_{ij}|^2}$. Then there are positive constants α, α_1, and α_2 (depending only on M and n) such that for any symmetric matrix A

$$||\bar{B}-A||_M \leq \{\sqrt{1-\alpha\theta^2} + \alpha_1||My-M^{-1}s||/||M^{-1}s||\}||B-A||_M$$
$$+ \alpha_2||y-As||/||M^{-1}s||$$

where $0 < \alpha \leq 1$ and
$\theta = ||M(B-A)s||/(||B-A||_M||M^{-1}s||)$ for $B \neq A$ with $\theta = 0$ otherwize.

Now the following lemma can be proved in a straightforward manner.

<u>Lemma 2.</u>
Assume that $F_{xx}(\bar{x},\bar{u}) > 0$ (this is the case if c is large enough) and that there exist constants K_1 and K_2 such that

$$||F_{xx}(x,u) - F_{xx}(\bar{x},\bar{u})|| \leq K_1||x-\bar{x}|| + K_2||u-\bar{u}|| \quad \text{for } (x,u) \text{ in}$$

some neighbourhood of $(\bar{x},\bar{u})$. Then if $x^{(k)}$ and $u^{(k)}$ converge to $\bar{x}$ and $\bar{u}$ there exists a k_0 such that the matrices $B^{(k)}$ generated by the formula A2 satisfy the following inequality for $k \geq k_0$. (Actually the lemma only requires that $B^{(k)}$ is generated by A2; $x^{(k)}$ and $u^{(k)}$ could be any sequences converging to $\bar{x}$ and $\bar{u}$)

$$||B^{(k+1)} - F_{xx}(\bar{x},\bar{u})||_M \leq \{\sqrt{1-\alpha\theta_k^2} + \alpha_3\sigma^{(k)}\}||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M + \alpha_4\sigma^{(k)}$$

where α_3 and α_4 are positive constants, $0 < \alpha \leq 1$ and

$$\sigma^{(k)} = \max(||x^{(k)} - \bar{x}||, ||x^{(k+1)} - \bar{x}||) + ||u^{(k+1)} - \bar{u}||$$

$$\theta^{(k)} = \frac{||M(B^{(k)} - F_{xx}(\bar{x},\bar{u}))s^{(k)}||}{||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M ||M^{-1}s^{(k)}||} \qquad B^{(k)} \neq F_{xx}(\bar{x},\bar{u})$$

$$\theta^{(k)} = 0 \qquad B^{(k)} = F_{xx}(\bar{x},\bar{u})$$

$$M = F_{xx}(\bar{x},\bar{u})^{-1/2}$$

Proof Taking $M=F_{xx}(\bar{x},\bar{u})^{-1/2}$ one gets

$$||My^{(k)} - M^{-1}s^{(k)}|| = ||M(y^{(k)} - F_{xx}(\bar{x},\bar{u}))s^{(k)}|| \leq$$

$$||M||^2 \{ K_1 \max(||x^{(k)} - \bar{x}||, ||x^{(k+1)} - \bar{x}||) +$$

$+ K_2||u^{(k+1)} - \bar{u}||\} \, ||M^{-1}s^{(k)}||$. Then there exists a k_0 such that

$$||My^{(k)} - M^{-1}s^{(k)}|| \leq \tfrac{1}{3}||M^{-1}s^{(k)}|| \qquad k \geq k_0$$

An application of Lemma 1 then gives the result.

It will now be shown that, provided the constant c appearing in $F(x,u)$ is large enough, and the starting point near enough to the optimum, the convergence will be at least linear.

Theorem 1

Take an r, $0 < r < 1$. Suppose that F_{xx} satisfies the conditions of Lemma 2. Then there exist constants $c_0 > 0$, $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ such that, if c is chosen greater than c_0 and if

$$||x^{(0)} - \bar{x}|| \leq \varepsilon_1 \quad \text{and} \quad ||B^{(0)} - F_{xx}(\bar{x},\bar{u})|| \leq \varepsilon_2$$

then the sequence $x^{(k)}$ generated by algorithm A2 satisfies

$$||x^{(k+1)} - \bar{x}|| \leq r||x^{(k)} - \bar{x}||$$

Proof.

Since h is continuously differentiable there exist constants K and ε_1' such that $||h(x)-h(\bar{x})|| \leq K||x-\bar{x}||$ for $||x-\bar{x}|| \leq \varepsilon_1'$. Let $\psi(u)$ denote the value of x which for the given value of u satisfies $F_x(\psi(u),u) = 0$. If only values of u and x sufficiently close to the solution are studied, this value is well defined, (Glad (1973)), and $\psi(u)$ is a continuously differentiable function. It then follows that

$$F_x^T(x^{(k)},u^{(k+1)}) = F_x^T(\psi(u^{(k+1)}),u^{(k+1)}) +$$

$$+ \int_0^1 F_{xx}(\theta(t),u^{(k+1)})(x^{(k)} - \psi(u^{(k+1)}))\, dt =$$

$$= \int_0^1 F_{xx}(\theta(t),u^{(k+1)})(x^{(k)} - \bar{x})\, dt -$$

$$- \int_0^1 F_{xx}(\theta(t),u^{(k+1)})(\psi(u^{(k+1)}) - \bar{x})\, dt$$

where $\theta(t) = \psi(u^{(k+1)}) + t(x^{(k)} - \psi(u^{(k+1)}))$

Now study the derivative of $\psi(u)$.

$$\psi_u(u) = -F_{xx}^{-1}(\psi(u),u)\, g_x^T(\psi(u)) =$$

$$= - \left(L_{xx}(\psi(u),u) + c g_x^T(\psi(u)) g_x(\psi(u)) \right)^{-1} g_x^T(\psi(u))$$

If c is large enough $L_{xx} + c g_x^T g_x$ is positive definite. Suppose that $c=c_1$ suffices and put $c=c_1+c_2$. Then

$$F_{xx} = L_{xx} + c_1 g_x^T g_x + c_2 g_x^T g_x = A + c_2 g_x^T g_x$$

where A is invertible. It then follows that

$$-\psi_u = \left(A + c_2 g_x^T g_x \right)^{-1} g_x^T =$$

$$= A^{-1}g_x^T - c_2 A^{-1} g_x^T (I + c_2 g_x A^{-1} g_x^T)^{-1} g_x A^{-1} g_x^T =$$

$$= \frac{1}{c_2} (g_x A^{-1} g_x^T)^{-1} + \mathcal{O}(\frac{1}{c_2^2})$$

Since all elements of ψ_u can be made arbitrarily small by choosing c large enough it is possible to find a c_0 such that, if $c \geq c_0$, then, for arbitrary $K_1 > 0$

$$||\psi(u^{(k+1)}) - \bar{x}|| = ||\psi(u^{(k+1)}) - \psi(\bar{u})|| \leq \frac{r}{2KK_1} ||u^{(k+1)} - \bar{u}|| =$$

$$= \frac{r}{2KK_1} ||h(x^{(k)}) - h(\bar{x})|| \leq \frac{r}{2K_1} ||x^{(k)} - \bar{x}|| \quad \text{if } ||x^{(k)} - \bar{x}|| \leq \varepsilon_1'$$

The following inequality is then true

$$||x^{(k+1)} - \bar{x}|| \leq ||I - (B^{(k)})^{-1} \int_0^1 F_{xx}(\theta(t), u^{(k+1)}) dt || \cdot || x^{(k)} - \bar{x}||$$

$$+ ||(B^{(k)})^{-1} \int_0^1 F_{xx}(\theta, t), u^{(k+1)}) dt || \cdot \frac{r}{2K_1} || x^{(k)} - \bar{x}||$$

From the conditions imposed on F_{xx} it then follows that there are constants ε_2' and ε_1'' such that

$$||I - (B^{(k)})^{-1} \int_0^1 F_{xx}(\theta(t), u^{(k+1)}) dt || \leq \frac{r}{2} \quad \text{if}$$

$$||B^{(k)} - F_{xx}(\bar{x}, \bar{u})|| \leq \varepsilon_2' \quad \text{and } ||x^{(k)} - \bar{x}|| \leq \varepsilon_1''.$$

Therefore, if the algorithm is started with $||x^{(0)} - \bar{x}|| \leq \varepsilon_1 = \min(\varepsilon_1', \varepsilon_1'')$ and $||B^{(0)} - F_{xx}(\bar{x}, \bar{u})|| \leq \varepsilon_2 < \varepsilon_2'$, then the inequality $||x^{(k+1)} - \bar{x}|| \leq r||x^{(k)} - \bar{x}||$ (1) will remain true as long as $||B^{(k)} - F_{xx}(\bar{x}, \bar{u})|| \leq \varepsilon_2'$.

Since $||u^{(k+1)} - \bar{u}|| \leq K||x^{(k)} - \bar{x}||$ the inequality of Lemma 2 can be written

$$||B^{(k+1)} - F_{xx}(\bar{x},\bar{u})||_M \leq (1 + \alpha_3' \delta^{(k)})||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M +$$

$$+ \alpha_4' \delta^{(k)} \qquad (2)$$

where $\delta^{(k)} = ||x^{(k)} - \bar{x}||$. Combining inequalities (1) and (2) it can be seen that , if $||B^{(0)} - F_{xx}(\bar{x},\bar{u})||_M$ and $||x^{(0)} - \bar{x}||$ are chosen small enough, then $||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M$ will never exceed ε_2' and (1) will therefore be true for all k, which proves the theorem.

It is now possible to use the following theorem analogous to a result by Dennis and Moré (1974).

<u>Theorem 2</u>.

Let $B^{(k)}$ satisfy the inequality

$$||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M \leq (\sqrt{1 - \alpha(\theta^{(k)})^2} + \alpha_3 \delta^{(k)})||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M$$

$$+ \alpha_4 \delta^{(k)}$$

where $\Sigma\ \delta^{(k)} < \infty$. Then

$$||(B^{(k)} - F_{xx}(\bar{x},\bar{u}))\,\hat{s}^{(k)}|| \to 0 \qquad k \to \infty$$

where $\hat{s}^{(k)} = s^{(k)}/||s^{(k)}||$

<u>Proof.</u>

See Dennis and Moré (1974).

<u>Corollary</u>

If the algorithm A2 is used with $||x^{(0)} - \bar{x}||$ and $||B^{(0)} - F_{xx}(\bar{x},\bar{u})||_M$ sufficiently small, then $||(B^{(k)} - F_{xx}(\bar{x},\bar{u}))\hat{s}^{(k)}|| \to 0$

Theorem 1 and the corollary to Theorem 2 show that the idea of updating u at every iteration seems to be reasonable, at least near the minimum. In the paper by Dennis and Moré the fact that $\|(B^{(k)} - F_{xx})\hat{s}^{(k)}\| \to 0$ is used to prove superlinear convergence for Quasi-Newton methods. It would be nice to prove superlinear convergence of algorithm A2 in a similar way. To do that it is necessary to specify more precisely what function h is. A natural choice is $h(x) = -(g_x g_x^T)^{-1} g_x f_x^T$, which is used by several authors, Miele et al. (1971), Fletcher (1970), Mårtensson (1972). This updating however gives only linear convergence as shown by the example in the appendix. A different choice is

$$h = h(x,B^{(k)}) = (g_x(B^{(k)})^{-1}g_x^T)^{-1}(g - g_x(B^{(k)})^{-1}f_x^T) - cg$$

which has been suggested by Pierson and O'Doherty (1974). This way of updating u can also be derived in the following way. The optimality conditions give the following equations

$$F_x(x,u) = 0$$

$$g(x) = 0$$

Solving these equations by a Newton-Raphson technique gives

$$F_{xx}\,\delta x + g_x^T\,\delta u = -F_x^T$$

$$g_x\,\delta x = -g$$

Replacing F_{xx} by $B^{(k)}$ and eliminating δx gives the formula for h shown above. It is also interesting to note that this way of updating u is a natural extension of the one used in Glad (1973). There u was updated after a complete minimization, making $F_x=0$, and for $F_x=0$ the two formulas agree.

Since the function h, in this case, depends on $B^{(k)}$ as well as x, the result of Theorem 1 can not be used directly. It is however possible to modify the theorem for this case. To do this note that $h(\bar{x},B^{(k)}) = \bar{u}$ for any positive definite matrix $B^{(k)}$. This follows since

$$h(\bar{x},B^{(k)}) = (g_x(\bar{x})(B^{(k)})^{-1}g_x^T(\bar{x}))^{-1}(-g_x(\bar{x})(B^{(k)})^{-1}f_x^T(\bar{x}))$$

$$= (g_x(\bar{x})(B^{(k)})^{-1}g_x^T(\bar{x}))^{-1}g_x(\bar{x})(B^{(k)})^{-1}g_x^T(\bar{x})\ \bar{u} = \bar{u}$$

where the fact that $g(\bar{x}) = 0$ and $f_x(\bar{x}) = -\bar{u}^T g_x(\bar{x})$ has been used. From the differentiability conditions of f_x and g_x it now follows that, if $||B^{(k)} - F_{xx}(\bar{x},\bar{u})||_M \leq \varepsilon_2'$ and $||x - \bar{x}|| \leq \varepsilon_1'$, there exists a constant K such that $||h(x,B^{(k)}) - h(\bar{x},B^{(k)})|| \leq K||x - \bar{x}||$. The proof can then be done according to Theorem 1. Consequently the following theorem is true.

<u>Theorem 3.</u>

Let $x^{(k)}$ and $u^{(k)}$ be generated by algorithm A2 with $h = h(x,B^{(k)}) = -(g_x(B^{(k)})^{-1}g_x^T)^{-1}(g - g_x(B^{(k)})^{-1}f_x^T) - cg$. Suppose that $||B^{(0)} - F_{xx}(\bar{x},\bar{u})|| \leq \varepsilon_1$, $||x^{(0)} - \bar{x}|| \leq \varepsilon_2$, where ε_1 and ε_2 are sufficiently small. Then there exists an r, $0<r<1$, such that $||x^{(k+1)} - \bar{x}|| \leq r||x^{(k)} - \bar{x}||$. The matrices $B^{(k)}$ satisfy $||(B^{(k)} - F_{xx}(\bar{x},\bar{u}))\hat{s}^{(k)}|| \to 0$, $k\to\infty$.

This result can be used to establish superlinear convergence of the algorithm A2 with $h = h(x,B^{(k)})$.

<u>Theorem 4.</u>

If algorithm A2 is used with $h = h(x,B^{(k)})$ defined above and if $||(B^{(k)} - F_{xx}(\bar{x},\bar{u}))\hat{s}^{(k)}|| \to 0$, then $x^{(k)}$ and $u^{(k)}$ converge superlinearly to $\bar{x}$ and $\bar{u}$ i.e.

$$\left\| \begin{pmatrix} x^{(k+1)} - \bar{x} \\ u^{(k+1)} - \bar{u} \end{pmatrix} \right\| \Bigg/ \left\| \begin{pmatrix} x^{(k)} - \bar{x} \\ u^{(k)} - \bar{u} \end{pmatrix} \right\| \to 0 \text{ as } k \to \infty .$$

Proof.

It is shown in Dennis and Moré (1974) that superlinear convergence of the iteration $x^{(k+1)} = x^{(k)} - (B^{(k)})^{-1} f(x^{(k)})$ for solving the equation $f(x)=0$ is equivalent to the condition

$$\| (B^{(k)} - f'(\bar{x}))(x^{(k+1)} - x^{(k)}) \| \,/\, \| x^{(k+1)} - x^{(k)} \| \to 0$$

if $f'(\bar{x})$ is nonsingular and f' continuous at $\bar{x}$ (f' is the derivative of f and $\bar{x}$ is the solution). Applying this result to the iteration

$$\begin{pmatrix} B^{(k)} & g_x^T \\ g_x & 0 \end{pmatrix} \begin{pmatrix} x^{(k+1)} - x^{(k)} \\ u^{(k+1)} - u^{(k)} \end{pmatrix} = - \begin{pmatrix} F_x^T \\ g \end{pmatrix}$$

shows that superlinear convergence is obtainded if

$$\frac{\left\| \left(\begin{pmatrix} B^{(k)} & g_x^T \\ g_x & 0 \end{pmatrix} - \begin{pmatrix} F_{xx}(\bar{x},\bar{u}) & g_x^T(\bar{x}) \\ g_x(\bar{x}) & 0 \end{pmatrix} \right) \begin{pmatrix} x^{(k+1)} - x^{(k)} \\ u^{(k+1)} - u^{(k)} \end{pmatrix} \right\|}{\left\| \begin{matrix} x^{(k+1)} - x^{(k)} \\ u^{(k+1)} - u^{(k)} \end{matrix} \right\|} \to 0 .$$

A sufficient condition for this is obviously that

$\| (B^{(k)} - F_{xx}(\bar{x},\bar{u})) \hat{s}^{(k)} \| \to 0.$

3. DISCUSSION OF GLOBAL CONVERGENCE:

In this section a general model for an algorithm which gives convergence for an arbitrary starting point will be given. To get some criterion for when an updating of u is good or not the vector

$$t(x,u) = \{L_{x_1},\ldots,L_{x_n}, g_1,\ldots g_p, |\min(-g_{p+1},u_{p+1})|\ldots, |\min(-g_m,u_m)|\}$$

is introduced. Since $t(x,u) = 0$ only when (x,u) satisfies the first order conditions for a Kuhn-Tucker point, t can be used to check convergence. The following algorithm is used.

Algorithm A3

0) Choose a starting point $x^{(0)},u^{(0)},B^{(0)},c^{(0)}$; put $i=0$, $j=0$, $k=0$; put $a_0 = ||t(x^{(0)},u^{(0)})||$

1) If $a_0 < \delta$ then stop

2) Compute a search direction $s^{(i)}$ from $B^{(i)}s^{(i)} = -F_x^T$

3) Perform a line search along $x = x^{(i)} + \alpha s^{(i)}$ such that the value of $F(x,u^{(i)})$ is reduced. This gives a new value $x = x^{(i+1)}$.

4) Update the matrix $B^{(i)}$ according to some Quasi-Newton formula.

5) Use some formula to compute a new value $\hat{u}$ for the multipliers and a vector $\hat{x}$ ($\hat{x}$ might be equal to $x^{(i+1)}$)

6) If $a = ||t(\hat{x},\hat{u})|| < \gamma a_0$, where $0<\gamma<1$, then put $x^{(k+1)} = \hat{x}$ $u^{(k+1)} = \hat{u}$, $i=i+1$, $a_0 = a$, $k=k+1$ and go to 1), else go to 7)

7) If $||F_x|| \leq \varepsilon^{(j)}$, then put $x^{(k+1)} = x^{(i)}$, $u^{(k+1)} = u^{(k)}$ and put $k=k+1$, $i=i+1$, else put $i=i+1$ and go to 1)

8) Put $c^{(j+1)} = Kc^{(j)}$ where $K>1$, put $j=j+1$ and go to 1)

The convergence properties of this model algorithm are given by the following theorem, which is analogous to a theorem in Polak (1971) dealing with penalty function methods.

Theorem 4.

Suppose that in algorithm A3 $\varepsilon^{(j)}\to 0$ and that for each x the vectors $\nabla g_i(x)$, $i\in\{1,..,p\}\cup\{i: i>p, g_i(x)\geq 0\}$, are linearly independent. Also assume that for fixed u and fixed c the algorithm will converge to a point where $F_x=0$. (This can be proved for several versions of the usual Quasi-Newton methods, provided the line search satisfies certain conditions, see Polak (1971)). Then every accumulation point of the sequence $x^{(k)}$ is a Kuhn-Tucker point.

Proof.

There are two cases. Either u is updated infinitely many times or else there exists a $k=k_1$ so that $u^{(k)}$ remains fixed for $k>k_1$. In the first case, since $||t||$ decreases with a factor γ each time, it is clear that if a subsequence converges to (x',u') say, then $t(x';u') = 0$, which means that (x',u') is a Kuhn-Tucker point. For the second case, notice that, since the unconstrained minimization method converges to a point where the derivative is zero, the condition $||F_x|| \leq \varepsilon^{(j)}$ will be satisfied infinitely many times. Now study a subsequence $(x^{(k_i)},u^{(k_i)})$ converging to (x',u'). To make the notation easier, the subsequence will be called just $(x^{(k)},u^{(k)})$ for the remainder of the proof. It is then true that

$$||f_x(x^{(k)}) + \sum_{j=1}^{p} (c^{(k)}g_j(x^{(k)})+u_j^{(k)})g_{jx}(x^{(k)}) +$$

$$+ \sum_{j=p+1}^{m} (c^{(k)}g_j(x^{(k)})+u_j^{(k)})_+ g_{jx}(x^{(k)})|| \leq \varepsilon^{(k)}$$

for sequences $c^{(k)}\to\infty$, $\varepsilon^{(k)}\to 0$.

If $g_j(x')<0$, then $c^{(k)}g_j(x^{(k)})+u_j^{(k)}< 0$ for k greater than some k_0. Therefore

$$||f_x(x^{(k)}) + c^{(k)}\{ \sum_{j=1}^{p} (g_j(x^{(k)}) + u_j^{(k)}/c^{(k)}) g_{jx}(x^{(k)}) +$$

$$+ \sum_{\substack{j=p+1\\ j\notin I'}}^{m} (g_j(x^{(k)}) + u_j^{(k)}/c^{(k)})_+ g_{jx}(x^{(k)})\}|| \leq \varepsilon^{(k)}$$

where $I' = \{j: j>p,\ g_j(x')<0\}$. Since the vectors $g_{jx}(x^{(k)})$, $j\notin I'$, are linearly independent for k greater than some k_2, it follows that $g_j(x^{(k)}) \to 0$, $j\notin I'$ and consequently x' satisfies $g_j(x') = 0$, $j=1,..,p$ and $g_j(x') \leq 0$ $j=p+1,..,m$. From the linear independence it also follows that $(c^{(k)}g_j(x^{(k)}) + u_j^{(k)}) \to \beta_j$, $j\notin I'$, where β_j satisfy $f_x(x') + \Sigma\ \beta_j g_{jx}(x') = 0$. Therefore x' is a Kuhn-Tucker point.

Let $D = \{$ x: x is a Kuhn-Tucker point$\}$. From the theorem it follows that if $x^{(k)}$ remains in some compact set for all k, then $x^{(k)} \to D$. It is difficult to give general conditions that guarantee that $x^{(k)}$ remains in some compact set. In practice however it can usually be achieved, if necessary by adding extra inequality constraints or modifying the function f.

4. PRESENTATION OF THE DIFFERENT ALGORITHMS

A few algorithms based on the ideas of the previous sections have been tested. They all require that the derivatives of the functions $f(x)$ and $g_i(x)$ are calculated. Both the methods of updating u discussed in section 2 are used:

(I) $u = -(g_x g_x^T)^{-1} g_x f_x^T$

(II) $u = (g_x B^{-1} g_x^T)^{-1}(g - g_x B^{-1} f_x^T) - cg$

The basic structures of the different algorithms are given below.

<u>Algorithm MINGR1</u>

0) Choose a starting point $x^{(0)}, u^{(0)}$, $B^{(0)}$, $c^{(0)}$ and put $i=0$. Compute $a_0 = ||r(x^{(0)}, u^{(o)}, c^{(0)})||^{*)}$. Put $N=1$.

1) If a_0 is sufficiently small then stop, else determine a search direction $s^{(i)}$ from
$B^{(i)} s^{(i)} = - F_x(x^{(i)}, u^{(i)})^T$

2) Perform a line search along $x = x^{(i)} + \alpha s^{(i)}$; determine an α which approximately minimizes $F(x^{(i)} + \alpha s^{(i)}, u^{(i)})$; call the new point $x^{(i+1)}$ and put $a_0 = \min(a_0, ||r(x^{(i+1)}, u^{(i)}, c^{(i)})||)$

3) Update $B^{(i)}$ using a Quasi-Newton formula.

4) If $||x^{(i+1)} - x^{(i)}|| \leq \varepsilon$ go to 9)

5) If i is a multiple of N, then update u using (II) and call the new value $\hat{u}$. Compute a new value of x from
$\hat{x} = x^{(i+1)} - (B^{(i+1)})^{-1} F_x(x^{(i+1)}, \hat{u})$.

6) Check if $||r(\hat{x}, \hat{u}, c^{(i)})|| \leq \beta a_0$; if true then put $x^{(i+1)} = \hat{x}$, $u^{(i+1)} = \hat{u}$, $N=1$, $i=i+1$, $a_0 = ||r||$ and go to 1)

*) $r(x,u,c) = (F_x(x,u), g_1(x), \ldots, g_p(x), |\min(u_{p+1}, -g_{p+1}(x))|, \ldots)$

else go to 7) (if an interpolation has already been done go to 8))

7) Calculate $\hat{x},\hat{u}$ from the interpolation formula $\hat{x}:= (1-\alpha)x^{(i+1)} + \alpha\hat{x}$, $\hat{u}:= (1-\alpha)u^{(i)} + \alpha\hat{u}$, where $\alpha = a_0/||r||$; go to 6)

8) Put $N = 2N$; put $u^{(i+1)}= u^{(i)}$, put $i=i+1$ and go to 1)

9) If the part of r containing the constraints is not sufficiently small then put $c_i= Kc_i$ $i=1,..,m$ and go to 1)

When the updating of the multipliers is not successful it is done more seldom because of use of the variable N.

Algorithm MINGR2

This algorithm is the same as MINGR1 except that the variable N is not used. Instead u is updated each time the gradient of F has decreased enough.

0)-4) Same as MINGR1

5) If $||F_x|| \leq \gamma b_0$ then update u and x using (II) and put $b_0=||F_x||$ else go to 1)

6)-9) Same as MINGR1

The following algorithms use the other method uf updating u.

Algorithm MINGR4

0) Choose a starting point $x^{(0)}$, $B^{(0)}$, $c^{(0)}$ and put $i=0$. Calculate u from (I) and compute $a_0=||t(x^{(0)},u^{(0)})||$ *).

1)-4) Same as MINGR1

5) Compute a vector $\hat{u}$ from (I). Put $\hat{x} = x^{(i+1)}$

6) Check if $||t(\hat{x},\hat{u})|| \leq \beta a_0$; if true then put $x^{(i+1)}= \hat{x}$, $u^{(i+1)}= \hat{u}$, $i=i+1$ and go to 1) else put $u^{(i+1)}= u^{(i)}$, put $i=i+1$ and go to 1)

7) Same as 9) in MINGR1.

Algorithm MINGR5

Same as MINGR4 except that every second time u is updated under 5) formula (II) is chosen for $\hat{u}$. $\hat{x}$ is then computed as under 5) of MINGR1.

*) $t(x,u)$ is defined as in section 3.

Algorithm MINGR7.

0) Choose a starting point $x^{(0)}, u^{(0)}$, $B^{(0)}$ and $c^{(0)}$ and put $i=0$. Compute $a_0 = ||t(x^{(0)}, u^{(0)})||$

1)-4) Same as MINGR1

5) Update x and u using (II) and the formula for x from 5) of MINGR1. Call the new values $\hat{x}$ and $\hat{u}$.

6) Check if $||t|| \leq \beta a_0$ and $F(\hat{x},\hat{u}) \leq F(x^{(i+1)},\hat{u})$; if true put $a_0 = ||t||$ and go to 7) else put $u^{(i+1)} = u^{(i)}$ and go to 1)

7) Update $B^{(i)}$ using a Quasi-Newton formula operating on $\hat{x} - x^{(i+1)}$ and $F_x(\hat{x},\hat{u}) - F_x(x^{(i+1)},\hat{u})$.

8) Put $x^{(i+1)} = \hat{x}$, $u^{(i+1)} = \hat{u}$, put $i=i+1$ and go to 5)

9) Same as MINGR1

For the inequality constraints, the formulas (I) and (II) are used in the following way. The vector g is replaced by $\tilde{g}$ defined by $\tilde{g} = (g_1(x),..,g_p(x),g_{i_1}(x),..,g_{i_q}(x))$, where $i_1,..$ $..,i_q$ are the indices for which $(c_i g_i(x) + u_i) > 0$. The multipliers u_i corresponding to the remaining indices are zeroed. If the updating formula gives a negative value for a multiplier corresponding to an inequality constraint, then that multiplier is zeroed. The linear minimization and the updating of the B-matrices in the algorithms above are based on the Quasi-Newton algorithm by Fletcher (1972). The parameters c_i are determined in the following way.

$$c_i = \frac{4 \max(1, (|f(x)-f(y)| + |f(x)-f(z)| + |f(z)-f(y)|))}{\sqrt{g_i^2(x) + g_i^2(y) + g_i^2(z)}}$$

where x,y and z are three different points. This means that all the algorithms above have the same values of c_i at the beginning. The normal values of $u^{(0)}$ and $B^{(0)}$ are 0 and the identity matrix respectively.

5. COMPARISON OF THE ALGORITHMS ON TEST PROBLEMS

To test the algorithms a number of test problems from the litterature are used. For comparison two other constrained minimization algorithms are also used, the one of Fletcher (1973), called VF01A in the Harwell library, and the GRG algorithm of Abadie (1970). The equality constrained problems are the following, which were also used in Glad (1973).

POW, see Powell (1969)

minimize $f(x) = x_1x_2x_3x_4x_5$

under the constraints

$g_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$

$g_2(x) = x_2x_3 - 5x_4x_5 = 0$

$g_3(x) = x_1^3 + x_2^3 + 1 = 0$

starting point $x = (-2,2,2,-1,-1)$

solution $\bar{x} = (-1.7171,1.5957,1.8272,-0.7636,-0.7636)$

PAV, see Himmelblau (1972)

minimize $f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

under the constraints

$g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$

$g_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$

starting point $x = (10,10,10)$

solution $\bar{x} = (3.512,0.217,3.552)$

EXP, see Himmelblau (1972)

Minimize $f(x) = \sum_{i=1}^{10} \{\exp(x_i)\ (c_i + x_i - \ln \sum_{j=1}^{10} \exp(x_j)\ )\}$

under the constraints

$g_1(x) = \exp(x_1) + 2\exp(x_2) + 2\exp(x_3) + \exp(x_6) + \exp(x_{10})$
$- 2 = 0$

$g_2(x) = \exp(x_4) + 2\exp(x_5) + \exp(x_6) + \exp(x_7) - 1 = 0$

$g_3(x) = \exp(x_3) + \exp(x_7) + \exp(x_8) + 2\exp(x_9) + \exp(x_{10})$
$- 1 = 0$

where $c_1 = -6.089$ $c_2 = -17.164$ $c_3 = -34.054$ $c_4 = -5.914$
$c_5 = -24.721$ $c_6 = -14.986$ $c_7 = -24.100$ $c_8 = -10.708$
$c_9 = -26.662$ $c_{10} = -22.179$

starting point $x_i = -2.3$ $i=1,..,10$
solution $\bar{x} = (-3.2,-1.9,-0.24,-\infty,-0.72,-\infty,-3.6,-4.0,-3.3,$
$-2.3)$

COL1, see Fletcher and Lill (1970) and Colville (1968)

Minimize $f(x) = \sum_{j=1}^{5} e_j x_j + \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3$

under the constraints

$g_1(x) = -3.5x_1 + 2x_3 + 0.25 = 0$

$g_2(x) = -9x_2 - 2x_3 + x_4 - 2.8x_5 + 4 = 0$

$g_3(x) = 2x_1 - 4x_3 + 1 = 0$

$g_4(x) = x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 - 5 = 0$

where the constants are given by

e_j	-15	-27	-36	-18	-12
c_{ij}	30	-20	-10	32	-10
	-20	39	-6	-31	32
	-10	-6	10	-6	-10
	32	-31	-6	39	-20
	-10	32	-10	-20	30
d_j	4	8	10	6	2

starting point $x = (0,0,0,0,1)$
solution $\bar{x} = (0.3000,0.3335,0.4000,0.4283,0.2240)$

TRIG

$$\text{Minimize} \quad f(x) = \sum_{i=1}^{n} (b_i E_i - f_i(x))^2$$

under the constraints

$g_i(x) = E_i - f_i(x) = 0 \quad i = 1,..,m$

where $b_i = 1 \quad i=m+1,..,n$

$b_i \neq 1 \quad i=1,..,m$

$$f_i(x) = \sum_{j=1}^{n} (A_{ij}\sin(x_j) + B_{ij}\cos(x_j))$$

$E_i = f_i(\bar{x})$ where $\bar{x}$ is the point chosen to be the minimum

A_{ij}, B_{ij}, E_i, b_i and $\bar{x}$ are given in the appendix.

The inequality constrained problems are

ROS, see Rosen and Suzuki (1965)

$$\text{Minimize} \quad f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

under the constraints

$$g_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0$$

$$g_2(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0$$

$$g_3(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \leq 0$$

starting point $x = (0,0,0,0)$ (ROSa)

$x = (3,3,3,3)$ (ROSb)

solution $\bar{x} = (0,1,2,-1)$

PROG, see Colville (1968)

$$\text{Minimize} \quad f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

under the constraints

$$0 \leqslant 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 -$$
$$- 0.0022053x_3x_5 \leqslant 92$$
$$90 \leqslant 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 +$$
$$+ 0.0021813x_3^2 \leqslant 110$$
$$20 \leqslant 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 +$$
$$+ 0.0019085x_3x_4 \leqslant 25$$
$$78 \leqslant x_1 \leqslant 102 \ , \quad 33 \leqslant x_2 \leqslant 45 \ , \quad 27 \leqslant x_3 \leqslant 45 \ ,$$
$$27 \leqslant x_4 \leqslant 45 \ , \quad 27 \leqslant x_5 \leqslant 45$$

starting point $x = (78,33,27,27,27)$

solution $\bar{x} = (78,33,29.99525562,45,36.77581389)$

The results are given in the following table. The entries are the number of times that (f,f_x,g,g_x) has been evaluated. For the GRG-method a separate table is presented because in this case f, f_x, g and g_x are not evaluated the same number of times. The first column gives the absolute accuracy in x which was required.

	acc	MINGR1	MINGR2	MINGR4	MINGR5	MINGR7	VF01A
POW	10^{-4}	22	35	37	34	18	39
PAV	10^{-3}	32	111	39	>150	35	135
EXP	10^{-1}	≥150	>150	95	>150	>150	140
COL1	10^{-4}	19	19	22	22	15	26
TRIG							
n=2	10^{-3}	12	10	18	11	11	14
n=4	10^{-3}	111	>150	32	32	21	>150
n=6	10^{-3}	52	53	36	23	25	63
n=8	10^{-2}	>150	>150	51	64	148	>150
ROSa	10^{-3}	22	41	23	84	26	-
ROSb	10^{-3}	28	30	>150	>150	20	-
PROG	10^{-3}	88	86	33	83	67	94

For TRIG n is the number of variables. The number of constraints was n/2. The sign "-" means that the algorithm was not tested on this problem. The computation was stopped after 150 function evaluations and if the desired accuracy was not reached then, the entry is marked ">150". For the problem PAV the algorithms found a local minimum different from the one given above: x = (0.332,4.678,-1.735).

Results for GRG. Number of evaluations to reach the accuracies given above.

	f	f_x	g	g_x
POW	82	33	101	15
PAV	114	26	192	11
EXP	192	73	611	27
COL1	26	17	22	7
TRIG				
n=2	24	8	29	6
n=4	121	29	230	14
n=6	201	59	564	33
n=8	235	75	902	29
PROG	67	30	131	12

It is clear that the comparison of GRG with the other algorithms depends heavily on the amount of work required to calculate the different functions and their gradients. The algorithms in the first table can be compared directly but since no algorithm is better than the others on all the problems there is no decisive result. It is interesting to note however that MINGR4 and MINGR7, the two algorithms closest to the model algorithms of sections 2 and 3, get quite good figures.

6. CONCLUSIONS

Section 3 shows that it is possible to construct algorithms, where the multipliers are updated at each iteration, which have at least as good global convergence properties as exteriour point penalty function methods. In section 2 it can be seen that the local convergence is good if u is updated in a suitable way. According to the results shown there, method II is preferable to method I (using the notation of section 4), since method II is superlinearly convergent while I only converges linearly. The numerical experiments of section 4 show however that it is not obvious which algorithm is best in practice. Perhaps a good method should switch between the methods according to a suitable criterion.

7. REFERENCES

Abadie, J: (1970)
Numerical Experiments with the GRG Method, in Abadie, J., ed.: Integer and Nonlinear Programming, North Holland Publishing Co, Amsterdam.

Bertsekas,D.P. (1973)
On Penalty and Multiplier Functions for Constrainded Minimization. EES Department Working Paper, Stanford University, California.

Colville, A.R. (1968)
A Comparative Study of Nonlinear Programming Codes, IBM New York Scientific Center Report 320-2949

Dennis, J.E. and Moré, J.J. (1974)
A Characterization of Superlinear Convergence and its Applications to Quasi-Newton Methods. Mathematics of Computation vol. 28, no.126, pp 549-560

Fletcher, R. (1970)
Methods for Nonlinear Programming. in Abadie, ed. : Integer and Nonlinear Programming, North Holland Publishing company

Fletcher, R. and Lill, S.A. (1970)
A Class of Methods for Nonlinear Programming II. Computational Experience. in Rosen,J.B., Mangasarian, O.L. and Ritter, K., ed.: Nonlinear Programming, Academic Press, London

Fletcher, R. (1972)
Fortran Subroutines for Minimization by Quasi-Newton Methods. Harwell Report AERE-R7125.

Fletcher, R. (1973)
An Ideal Penalty Function for Constrained Optimization. Harwell Report C.S.S.2.

Glad, T. (1973)
Lagrange Multiplier Methods for Minimization under Equality Constraints. Report 7323, Lund Institute of Technology, Division of Automatic Control.

Himmelblau, D.M. (1972)
Applied Nonlinear Programming. McGraw Hill, New York.

Miele, A., Cragg, E.E., Iyer, R.R. and Levy, A.V. (1971)
Use of the Augmented Penalty Function in Mathematical Programming Problems. J. Opt. Theory Appl., vol.8, no.2, pp 115-130, pp 131-153.

Mårtensson, K. (1972)
New Approaches to the Numerical Solution of Optimal Control Problems. Report 7206, Lund Institute of Technology, Division of Automatic Control.

Pierson, B.L. and O'Doherty, R.J. (1974)
A Numerical Study of Multiplier Methods for Constrained parameter Optimization. Inst. J. Systems Sci. vol.5,no.2, pp 187-200.

Polak, E. (1971)
Computational Methods in Optimization. Academic Press, New York and London.

Powell, M.J.D. (1969)
A Method for Nonlinear Constraints in Minimization Problems. in Fletcher, R., ed.: Optimization, Academic Press London.

Rosen, J.B. and Suzuki, S. (1965)
Construction of Nonlinear Programming Test Problems, Comm. ACM, vol.8 p 113.

Tripati, S.S. and Narendra, K.S. (1972)
Constrained Optimization Problems Using Multiplier Methods, J. Opt. Theory Appl., vol.9, no.1, pp59-70.

8. APPENDIX

Example where $u = -(g_x g_x^T)^{-1} g_x f_x^T$ gives only linear convergence,

minimize $f(x)$ under the constraint $g(x)$

$f(x) = 0.5(x_1^2 + x_2^2)$ $\qquad g(x) = x_1 - 1$

Take $x^{(0)} = (0,0)^T$

$$B^{(0)} = \begin{pmatrix} 1+c & 0 \\ 0 & 1 \end{pmatrix}$$

Then $u^{(k+1)} = -x_1^{(k)}$, $\quad B^{(k)} = B^{(0)}$, $\quad x_2^{(k)} = 0$

$x_1^{(k+1)} - 1 = (x_1^{(k)} - 1)/(c+1)$

Coefficients in TRIG

```
TRIGONOMETRIC FUNCTION WITH N=  2 ,M=  1
******************************

A=
    .72479248-02    -.49803540+02
    .52957930+02    -.62673827+02

B=
  -.49956511+02     .78773880+00
    .60657923+02    .28036484+02

E=
  -.68468137+02     .39746146+02

b
    .82282977+00    .10000000+01

x̄
    .10000000+01    .10000000+01



TRIGONOMETRIC FUNCTION WITH N=  4 ,M=  2
******************************

A=


    .41369501+02    .73696531+02    .25613161+02   -.58377710+02
  -.25460052+02   -.29681022+02   -.21992548+02   -.91697049+02
    .30854866+02   -.67154062+02    .41992146+02   -.46604071+01
    .64758375+02   -.56855083+02    .31184877+02   -.33408517+02

B=
  -.92329782+02   -.26853132+02    .45356791+02   -.58476639+02
    .23529812+02   -.39853686+02   -.73271728+02    .92636328+01
  -.93243503+02   -.18732467+02   -.79455309+02   -.62864277+02
    .43283844+01   -.80085210+02   -.42124230+02   -.21332652+02

E=
  -.22291780+01   -.18546966+03   -.13922913+03   -.70438223+02

b
    .22681122+00   -.71919782+00    .10000000+01    .10000000+01
x̄
    .10000000+01    .10000000+01    .10000000+01    .10000000+01
```

```
TRIGONOMETRIC FUNCTION WITH N=  6 ,M=  3
******************************

A=
  -.85648038+02   -.28826805+02    .68610128+02   -.30045291+02
  -.14296383+02    .37462990+02    .32342091+02   -.52344628+02
   .24953208+02    .27274352+02   -.80266293+02    .95992870+02
   .95906511+02    .14593713+02   -.55736202+02   -.85337452+02
   .13183112+02   -.58740739+02    .24844017+02    .51933079+01
   .89147961+02   -.37819559+02   -.17327652+01    .82195255+02

      .17683649+01    .15496438+02
     -.61907854+02    .25578873+02
     -.20561994+02   -.45534253+02
      .78564411+02   -.73505751+02
      .81114780+02    .89413368+02
     -.30134047+02   -.50219104+02

B=
   .83390203+02   -.73471916+02    .72907646+02    .13560187+02
  -.98534141+02    .61584833+02    .39789032+02   -.72168315+02
  -.58024588+02   -.14132978+02    .95599434+02    .65562305+02
  -.53708071+02    .70934534+02   -.22827277+02   -.65788393+01
  -.23348913+02   -.42303835+02    .29798244+02   -.87024365+02
   .69085348+02    .13152628+01    .77766790+02   -.56729583+02

     -.11431505+02   -.41381979+01
      .28068092+02   -.49139615+02
      .36568027+02   -.23170252+01
      .30595661+02    .33605288+02
     -.80092393+02    .57311378+02
     -.20237652+02   -.69175381+02

E=
  -.56829398+01   -.76750702+02    .68158418+02    .66372886+01

      .51734235+02    .44377367+02

b
   .36920038+00    .94098282+00   -.67690657+00    .10000000+01

      .10000000+01    .10000000+01

x̄
   .10000000+01    .10000000+01    .10000000+01    .10000000+01

      .10000000+01    .10000000+01
```

TRIGONOMETRIC FUNCTION WITH N= 8 ,M= 4
****

A=

.46972275+02	.73544701+02	-.80949702+02	.85789461+02
.96981817+02	-.22059572+02	.47405386+02	.40123520+02
-.17271843+02	.14967769+02	.18436436+02	-.30528218+02
.76226749+02	.92170975+02	.84711514+02	-.91036454+02
-.12619244+02	-.41539893+02	-.25553151+02	.47800207+01
.83357851+02	.83558083+02	.20647049+01	-.81035419+02
.19745394+02	-.11987948+02	-.65151297+02	.48301329+02
.66334469+02	.47476391+02	-.68511724+02	.71209957+02

-.98862698+02	-.28470101+02	-.45779991+01	.73675707+02
.82387785+02	.82980247+02	.70238379+02	-.57109328+02
-.92849747+02	-.98503641+02	.47769365+02	-.11359116+02
.49720600+02	.68925529+02	.63295473+02	-.43649445+02
-.94157072+02	.68000397+02	.50731911+02	.55146770+02
.74125519+02	.48124857+02	.62081095+02	.69350744+02
.96675627+02	.27754297+02	-.81143924+02	-.86846743+01
-.18774525+02	-.59431486+01	-.86234849+02	-.20830666+02

B=

.91049196+02	.71825436+02	-.82126372+02	.53873743+02
.91796131+02	-.58521858+02	.61128666+02	-.94168806+01
.93253593+02	-.99475732+02	-.44100175+02	-.86267349+02
.21368621+02	-.89291743+02	-.38105596+02	-.32679787+01
-.67518912+02	-.41568763+02	.20800340+02	-.58522939+02
-.69370333+02	-.24318885+02	.31257820+02	.82467854+02
-.72380026+02	.79492922+02	.93656670+02	.96897932+02
-.92523546+01	-.31870826+02	.25767469+02	.45352516+02

-.28039126+02	.81531912+02	-.61254444+02	.43343863+02
-.20921739+02	.36176769+02	.95839350+02	-.52093630+01
-.80691957+02	.35206160+02	-.30239263+02	-.96000944+02
-.72264972+02	.13937540+02	.54334969+02	.49089355+02
-.38235605+02	-.47877544+02	.85288994+02	-.36720325+02
.25424223+01	-.34132655+02	.79680101+02	-.51016443+02
-.42027635+02	.94774126+02	.60170063+02	.56362129+02
-.70819041+02	.17128639+01	-.32824115+02	.20433415+02

E=

.14844264+03	.39002613+03	-.30907751+03	.21806108+03
-.95576873+02	.29671407+03	.21972678+03	-.40678316+02

b

-.89923885+00	.26556693+00	-.81345238+00	.72919089+00
.10000000+01	.10000000+01	.10000000+01	.10000000+01

$\bar{x}$

.10000000+01	.10000000+01	.10000000+01	.10000000+01
.10000000+01	.10000000+01	.10000000+01	.10000000+01