ON THE GLOBAL STABILIZATION OF LOCALLY CONVERGENT ALGORITHMS FOR OPTIMIZATION AND ROOT FINDING

by

E. Polak

Memorandum No. ERL-M491

29 October 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

On the Global Stabilization of Locally Convergent Algorithms for Optimization and Root Finding

by

E. Polak

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

Abstract

There are a number of algorithms in the literature which both theoretically and empirically are known to be only locally convergent. These include such well known algorithms as secant, Newton, quasi-Newton and primal-dual algorithms. Locally, these algorithms tend to be highly efficient. Consequently, it is very desirable to find ways of extending, or modifying, these algorithms, so that they become globally convergent while retaining their attractive local properties. This paper describes a set of techniques which have recently emerged for stabilizing such algorithms and illustrates their application by means of a number of examples.

Research sponsored by the National Science Foundation Grant GK-37672 and the U. S. Army Research Office -- Durham Contract DAHC04-73-C-0025.

1. Introduction.

As one examines the literature on optimization and solution of equations, one comes across a number of locally convergent algorithms with very attractive properties. In this category, we find such well known algorithms as the classical Newton method [7] the more recent extensions of Newton's method to the solution of simultaneous, nonsquare, systems of equations and inequalities [13], [14], various versions of secant and quasi-Newton methods [2], [3], [4], and multiplier methods [6], [12], [5], [1]. The first set of these algorithms converge superlinearly, provided a sufficiently good initial guess is available, while the methods of multipliers reduce a constrained optimization problem to an unconstrained one and converge, some superlinearly, provided a suitable convexification parameter has been found.

Over the last few years, a number of papers have been written such as [7a] [8] [9] [11], in which these locally convergent algorithms have been stabilized, i.e., made globally convergent, without loss of their local properties. An examination of these papers reveals a certain pattern in the manner in which the stabilization has been accomplished. In this paper we shall formalize this pattern into a set of theorems which we shall then illustrate by examples from the literature.

2. Abstract Stabilization Schemes.

We begin by stating an abstract problem.

Problem 1: Given a closed set F of feasible points in a Banach space X, find a point $z^* \in \Delta \subset F$, where Δ, a closed set, is the set of desirable points.

As a reference point, we need the following algorithm model (1.3.9 in [10]) for solving Problem 1, with A: $F \to 2^F$ an iteration map and c: $F \to \mathbb{R}^1$ an abstract cost.

Algorithm Model 1.

Data: $z_0 \in F$.

Step 1: Compute $y \in A(z_i)$.

Step 3: If $c(y) < c(z_i)$, set $z_{i+1} = y$, $i = i+1$ and go to step 1; else, stop. ¤

We shall also need the following result (Theorem (1.3.10) in

Theorem 1: Suppose (i) $c(\cdot)$ is continuous and (ii) for every $z \in F$, $z \notin \Delta$, there exist $\varepsilon(z) > 0$ and $\delta(z) < 0$ such that

2.1. $c(z'') - c(z') \leq \delta(z) \;\forall\, z' \in B(z,\varepsilon(z)),\; \forall\, z'' \in A(z')$,

where

2.2. $B(z,\varepsilon(z)) \triangleq \{z' \,|\, \|z'-z\| \leq \varepsilon(z)\}$

Under these assumptions, (i), if Algorithm Model 1 constructs an infinite sequence $\{z_i\}_{i=0}^{\infty}$, then every accumulation point is in Δ; (ii) if $\{z_i\}$ is finite, then its last element is in Δ. ¤

Our first, and simplest, estension of Algorithm Model 1 is directed to the stabilization of a locally convergent algorithm of the form $z_{i+1} \in A_1(z_i)$, $i = 0, 1, 2, \ldots$, where A_1: $F_1 \to 2^F$, with $\Delta \subset F_1 \subset F$. We assume that in addition to A_1 we also have an iteration map A_2: $F \to 2^F$ and an abstract cost function c: $F \to \mathbb{R}^1$, both satisfying the assumptions of Theorem 1 (with $A = A_2$).

Algorithm Model 2

Data: $z_0 \in F$.

Step 0: Set $i = 0$.

Step 1: If $z_i \in F_1$ compute a $y \in A_1(z_i)$ and go to step 2; else compute a $y \in A_2(z_i)$ and go to step 2.

Step 2: If $c(y) < c(z_i)$, set $z_{i+1} = y$ and go to step 1; else stop. ¤

Theorem 2: (i) Suppose that the map A: $F \to 2^F$ defined by $A(z) = A_1(z)$ if $z \in F_1$, $A(z) = A_2(z)$ otherwise, together with $c(\cdot)$ satisfy the assumptions of Theorem 1, then the conclusions of Theorem 1 apply to Algorithm Model 2. (ii) In addition, suppose there exists an open subset F_Δ, whose closure $\overline{F}_\Delta \subset F_1$, such that $\Delta \subset F_\Delta$, and $A_1(\overline{F}_\Delta) \subset \overline{F}_\Delta$, and that Algorithm Model 2 constructs a compact infinite sequence $\{z_i\}_{i=0}^{\infty}$. Then there exists an integer N such that $z_{i+1} \in A_1(z_i)$ for all $i \geq N$.

Proof: Part (i) is obvious. (ii) Since by (i) every accumulation point of $\{z_i\}$ is in Δ, and $\Delta \subset F_\Delta$, with F_Δ an open set, there exist an N such that $z_N \in F_\Delta$. But this implies that $z_i \in \overline{F}_\Delta$ for all $i \geq N$. ¤

Our next model is significantly different and is directed to the stabilization of superlinearly converging algorithms of the form $x_{i+1} \in A_1(x_i)$, $i = 0, 1, 2, \ldots$, where A_1: $F_1 \to 2^F$, with $\Delta \subset F_1 \subset F$, with an associated abstract cost c_1: $F_1 \to \mathbb{R}^+$. Again we assume that we have a map A_2: $F \to 2^F$ and a c_2: $F \to \mathbb{R}^1$, satisfying the assumptions of Theorem 1 (with $c = c_2$, $A = A_2$).

Algorithm Model 3

Data: $\gamma \in (0,1)$, $x_0 \in F$.

Step 0: Set $i = 0$, $j = 0$.

Step 1: If $x_i \in F_1$ and $c_1(x_i) \leq \gamma^j$ go to step 2; else go to step 3.

Step 2: Compute an $x_{i+1} \in A_1(x_i)$, set $z_j = x_i$, set $j = j+1$, set $i = i+1$, and go to step 1.

Step 3: Compute a $y \in A_2(x_i)$.

Step 4: If $c_2(y) < c_2(x_i)$, set $x_{i+1} = y$, $i = i+1$ and go to step 1; else, stop. ¤

Theorem 3: Suppose (i) $A_2(\cdot)$, $c_2(\cdot)$ satisfy the assumptions of Theorem 1; (ii) $c_1(\cdot)$ is continuous; (iii) $\forall\ \gamma \in (0,1)$ there exist an open set $F_{\gamma\Delta}$ such that $\Delta \subset F_{\gamma\Delta} \subset \overline{F}_{\gamma\Delta} \subset F_1$, $A_1(F_{\gamma\Delta}) \subset F_{\gamma\Delta}$ and

2.3. $\quad c_1(x') \leq \gamma\ c_1(x) \qquad \forall\ x \in F_{\gamma\Delta}, \quad \forall\ x' \in A_1(x).$

(iv) if $c_1(x^*) = 0$, then $x^* \in \Delta$.

Under these assumptions, (i) if Algorithm Model 3 constructs a compact infinite sequence $\{x_i\}$, then every accumulation point of $\{x_i\}$ is in Δ; (ii) if $\{x_i\}$ is finite, then its last element is in Δ; (iii) if $\{x_i\}$ is infinite and compact, then there exists an integer N such that $x_{i+1} \in A_1(x_i)$ for all $i \geq N$.

Proof: (ii) is obvious. Hence we only need to prove (i) and (iii).

(i) (a) Suppose that there is an N' such that $x_{i+1} \in A_2(x_i)$ for all $i \geq N'$. Then by Theorem 1, all accumulation points of $\{x_i\}$ are in Δ.

(b) Suppose that there exists an infinite subset K of $\{0,1,2,\ldots\ldots\}$ such that $x_{i+1} \in A_1(x_i)$ for $i \in K$. Then we must have $c_1(z_j) \to 0$ as $j \to \infty$. Since $\{z_j\}$ is a subsequence of $\{x_i\}$ and $\{x_i\}$ is compact, there exist a $z^* \in F$, and a subsequence $\{z_j\}_{j \in K_1}$ such that $z_j \xrightarrow{K_1} z^*$ and hence we must have $c_1(z^*) = 0$, which implies that $z^* \in \Delta$. But $F_{\gamma\Delta}$ is an open set and

$z_j \xrightarrow{K_1} z^*$. Hence there exist an N and a j_N such that $z_{j_N} = x_N \in F_{\gamma\Delta}$. Now $c_1(z_{j_N}) \leq \gamma^{j_N}$, by construction of z_{j_N}; it now follows from the fact that $A_1(F_{\gamma\Delta}) \subset F_{\gamma\Delta}$, that $x_{i+1} \in A_1(x_i)$ for all $i \geq N$, $x_i \in F_{\gamma\Delta}$ for all $i \geq N$, and $c_1(x_i) \to 0$ as $i \to \infty$. Consequently, any accumulation point x^* of $\{x_i\}$ satisfies $c_1(x^*) = 0$, i.e. $x^* \in \Delta$.

(iii) To show that (a) above cannot occur, suppose $j_{N'}$ is the last value of j. Hence, since $\{x_i\}$ is compact, and by (i) it has accumulation points $x^* \in \Delta$, there must exist and N such that $x_N \in F_{\gamma\Delta}$ and $c_1(x_N) \leq \gamma^{j_{N'}}$. But this immedaitely implies that $x_i \in F_{\gamma\Delta}$ for all $i \geq N$, i.e., that $x_{i+1} \in A_1(x_i)$ for all $i \geq N$. ¤

We now turn to the stabilization of algorithms whose convergence depends upon the correct choice of a parameter and there is no practical a priori way for selecting this parameter. We model a scheme for automatically selecting the required parameter, as follows. We assume we are given sequences of iteration maps A_j: $F \to 2^F$, of abstract costs c_j: $F \to \mathbb{R}^1$, of tests t_j: $F \to \mathbb{R}^1$ and subsets $\Delta_j \subset F$, with $j = 0, 1, 2, \ldots$.

<u>Algorithm Model 4</u>

<u>Data</u>: $x_0 \in F$.

<u>Step 0</u>: Set $i = 0$, $j = 0$.

<u>Step 1</u>: If $t_j(x_i) \leq 0$, go to step 2; else go to step 4.

<u>Step 2</u>: Compute a $y \in A_j(x_i)$.

<u>Step 3</u>: If $c_j(y) < c_j(x_i)$ set $x_{i+1} = y$, $i = i+1$ and go to step 1; else stop.

<u>Step 4</u>: Set $z_j = x_i$, set $j = j+1$ and go to step 1. ¤

<u>Theorem 4</u>: Suppose that (i) for each j, $j = 0, 1, 2 \ldots$, c_j, A_j, Δ_j satisfy the assumptions of Theorem 1, (with c, A, Δ replaced by c_j, A_j,

Δ_j); (ii) $t_j(\cdot)$ is continuous for $j = 0, 1, 2, \ldots$, (iii) for $j = 0, 1, 2, \ldots$, $\{z \in \Delta_j | t_j(z) \leq 0\} \subset \Delta$; (iv) for every $z^* \in F$ there exists a j^* and an $\varepsilon^* > 0$ such that $t_j(z) \leq 0$ for all $j \geq j^*$, for all $z \in B(z^*,\varepsilon^*)$. Under these assumptions, (i) if Algorithm Model 4 constructs a finite sequence $\{z_j\}$, and $\{x_i\}$ is infinite, then every accumulation point of $\{x_i\}$ is in Δ; (ii) if $\{z_i\}$ is finite and $\{x_i\}$ is finite, then the last element of $\{x_i\}$ is in Δ; (iii) if $\{z_j\}$ is infinite, then $\{z_j\}$ has no accumulation points.

Proof: Again (ii) is obvious and we only need to prove (i) and (iii). (i) Suppose that $\{z_j\}$ is finite, with last element $z_{j'}$. Then, by Theorem 1, all accumulation points of $\{x_i\}$ are in $\Delta_{j'}$. Furthermore, since $t_{j'}(\cdot)$ is continuous and $t_{j'}(x_i) \leq 0$ for all $i \geq i'$, for some finite i', if x^* is an accumulation point of $\{x_i\}$ it must also satisfy $t_{j'}(x^*) \leq 0$. Hence $x^* \in \Delta$. (iii) Suppose $\{z_j\}$ is infinite and that it has an accumulation point z^*. Then there exists an infinite subsequence $\{z_j\}_{j \in K}$ such that $z_j \xrightarrow{K} z^*$. Now, by assumption, there exist integers j^* and N such that $t_k(z_j) \leq 0$ for all $k \geq j^*, j \geq N$, $j \in K$. Hence, given any $j \in K$, $j \geq \max\{N,j^*\}$, we find that the Algorithm Model 4 would not increment j to $j+1$ at $x_i = z_j$, which is a contradiction. Hence $\{z_i\}$ has no accumulation points. ¤

3. Applications

We present applications without proof. Those interested, will find the proofs in the corresponding references. Our first application is to the stabilization of Newton's method for the problem

3.1 $$\min\{f^0(z) | z \in \mathbb{R}^n\}$$

where $f^0(\cdot)$ is three times continuously differentiable, but not necessarily convex. For this case, we introduce three parameters: $\alpha \in (0,1)$, $\beta \in (0,1)$, and $\gamma > 0$ large, and the notation $H(z) \equiv \frac{\partial^2 f^0(z)}{\partial z^2}$. We now define $F = \mathbb{R}^n$, $c(\cdot) = f^0(\cdot)$,

3.2a $$\Delta = \{z | \nabla f^0(z) = 0\}$$

3.2b $$F_1 = \{z \in \mathbb{R}^n | \|H(z)^{-1}\|_\infty \leq \gamma,\ f^0(z-H(z)^{-1}\nabla f(z)) - f^0(z) \leq -\alpha \langle \nabla f^0(z), H(z)^{-1}\nabla f^0(z) \rangle \}$$

and A_1: $F_1 \to \mathbb{R}^n$ by

3.3 $$A_1(z) = z - H(z)^{-1}\nabla f^0(z)$$

and A_2: $\mathbb{R}^n \to \mathbb{R}^n$ by

3.4 $$A_2(z) = z - \beta^{k(z)}\nabla f^0(z)$$

where (with $\mathbb{Z} = \{0,1,2,3,\ldots .\}$)

3.5 $$k(z) = \arg\min\{k \in \mathbb{Z} | f^0(z-\beta^k\nabla f^0(z)) - f^0(z) \leq -\beta^k\alpha\|\nabla f^0(z)\|^2\}$$

The stabilized algorithm is obtained by direct substitution into Algorithm Model 2 (see [9] for a more complicated application).

Next we consider an application of Algorithm Model 3 to the stabilization of the Pshenichnyi-Robinson [13], [14] extension of Newton's algorithm (see [11]).

This algorithm solves problems of the form: find $z \in \Delta \subset \mathbb{R}^n$ with

3.6 $$\Delta \triangleq \{z | g(z) = 0,\ f(z) \leqq 0\},$$

where g: $\mathbb{R}^n \to \mathbb{R}^m$, f: $\mathbb{R}^n \to \mathbb{R}^q$ are continuously differentiable functions. We use superscripts to denote components of f, g, z, etc.

After selecting some b >> 1 we define, $\bar{f}$: $\mathbb{R}^n \to \mathbb{R}^{q+1}$ by

3.7 $$\bar{f}^j(z) \triangleq \begin{cases} f^j(z) & j = 1,\ldots,q \\ \|z\|^2 - b & j = q+1 \end{cases}$$

3.8 $$G(z) \triangleq \frac{\partial g(z)}{\partial z}, \qquad \bar{F}(z) \triangleq \frac{\partial \bar{f}(z)}{\partial z}$$

3.9 $$\bar{f}^{j+}(z) \triangleq \max\{0, \bar{f}^j(z)\},\ j = 1,2,\ldots,q+1,$$

Next we define $F = \mathbb{R}^n$, and the cost function c_2: $\mathbb{R}^n \to \mathbb{R}^1$ by

3.10 $$c_2(z) \triangleq \frac{1}{2}\|g(z)\|^2 + \frac{1}{2}\|\bar{f}^+(z)\|^2$$

It is not difficult to see that $c_2(\cdot)$ is continuously differentiable. Using a given initial guess, z_0, we define

3.11 $$C(z_0) = \{z \mid c_2(z) \le c_2(z_0)\}$$

The functions A_1, A_2 and c_1 and the set F_1 will be defined in the algorithm; but first we state assumptions which ensure that our algorithm is globally convergent.

Assumption 3.1: The derivative matrices $G(\cdot)$ and $F(\cdot)$ are Lipschitz continuous.

Assumption 3.2: The pair $(\bar{F}(z), G(z))$ satisfies the Robinson LI condition [14] for all $z \in C(z_0)$, where z_0 is the initial guess to a solution for (1) and b is sufficiently large to ensure that the set $C(z_0)$ contains at least one such solution; i.e. for all $z \in C(z_0)$

$$u^T\bar{F}(z) + v^T G(z) = 0$$

and $u \geqq 0$, implies that $u = 0$ and $v = 0$. ¤

Algorithm

Data: $z_0 \in \mathbb{R}^n$, $b >> \|z_0\|^2$, $\alpha \in (0,1/2)$, $\beta \in (0,1)$, $\hat{\ell} > 1$, $\gamma \in (0,1)$.

Step 0: Set $i = 0$, $j = 0$, $x_0 = z_0$.

Step 1: Compute $g(x_i)$, $\bar{f}(x_i)$, $c_2(x_i)$, $G(x_i)$, and $\bar{F}(x_i)$. Stop if $c_2(x_i) = 0$.

Step 2: Compute a vector v_i which solves the problem

3.12 $$c_1(x_i) \triangleq \min\{\|v\|_\infty | g(x_i) + G(x_i)v = 0,\ \bar{f}(x_i) + \bar{F}(x_i)v \leqq 0\}$$

where $\|v\|_\infty \triangleq \max_r |v^r|$

Comment: Due to Assumption 3.2, the linearized problem (3.12) always has a solution, obtainable by linear programming techniques.

Step 3: If $c_1(x_i) \leq \gamma^j$ and $x_i + v_i \in C(z_0)$, set $x_{i+1} = x_i + v_i$, set $z_j = x_i$, set $j = j + 1$, set $i = i + 1$ and go to step 1; else go to step 4.

Note: $F_1 = \{x_i \in C(z_0) | x_i + v_i \in C(z_0)\}$, $A_1(x_i) = \{x_i + v_i | v_i \in \text{Arg}\min\{\|v\|_\infty | g(x_i) + G(x_i)v = 0,\ \bar{f}(x_i) + \bar{F}(x_i)v \leq 0\}$. Steps 2, 4, 5, 6, 7, 8 define a map $A_2': F_2 \to 2^{\mathbb{R}^n}$, where $F_2 \subset \mathbb{R}^n$ and steps 9, 10, 11, 12 define a map $A_2'': \mathbb{R}^n \to 2^{\mathbb{R}^n}$.

Step 4: Set $w_i = v_i$, $\phi(x_i) = -\dot{c}_2(z_i)$.

Step 5: Set $\ell = 0$.

Step 6: Compute $c_2(x_i + \beta^{\ell} w_i)$.

Step 7: If

$$3.1 \qquad c_2(x_i + \beta^{\ell} w_i) - c_2(x_i) \leq \beta^{\ell} \alpha \phi(x_i)$$

set $\ell_i = \ell$, set $x_{i+1} = x_i + \beta^{\ell_i} w_i \in A_2'(x_i)$, set $i = i + 1$ and go to step 13; else go to step 8.

Step 8: If $\ell < \hat{\ell}$ set $\ell = \ell + 1$ and go to step 6; else go to step 9.

Step 9: Compute $\nabla c_2(x_i)$, set $w_i = -\nabla c_2(x_i)$ and set $\phi(x_i) = -\|\nabla c_2(x_i)\|^2$.

Step 10: Set $\ell = 0$.

Step 11: Compute $c_2(x_i + \beta^{\ell} w_i)$.

Step 12: If (3.13) is satisfied, set $\ell_i = \ell$, set $x_{i+1} = x_i + \beta^{\ell_i} w_i \in A_2''(x_i)$ set $i = i + 1$ and go to step 1; else set $\ell = \ell + 1$ and go to step 11. ¤

Thus, in the algorithm above, $A_2(x) = A_2'(x)$ if (3.13) is satisfied at x, otherwise $A_2(x) = A_2''(x)$.

Finally, we present an algorithm [8] which corresponds to Algorithm Model 4. This algorithm is of the primal-dual type and it solves problems of the form

$$3.14 \qquad \min\{f^0(z) \mid g(z) = 0\}$$

where f: $\mathbb{R}^n \to \mathbb{R}^1$ and g: $\mathbb{R}^n \to \mathbb{R}^m$, with $m \leq n$, are three times continuous differentiable and, for any $z \in \mathbb{R}^n$ the matrix $\frac{\partial g(z)}{\partial z}$ has maximum rank. This algorithm uses a family of convexified Laprangians

c_j: $\mathbb{R}^n \to \mathbb{R}^1$, defined (as by Fletcher [5]) by

3.15 $$c_j(x) = f^0(x) + \langle y(x), g(x) \rangle + \frac{1}{2}\gamma_j \|g(x)\|^2, \; j = 0, 1, 2...$$

where for $j = 0, 1, 2...$ $\gamma_{j+1} > \gamma_j > 0$, and $\gamma_j \to \infty$ as $j \to \infty$ is a pre-selected sequence and

3.16 $$y(x) = -\left(\frac{\partial g(x)}{\partial x}\frac{\partial g(x)}{\partial x}^T\right)^{-1}\frac{\partial g(x)}{\partial x}\nabla f(x).$$

The algorithm uses a Gauss-Newton type approximation $H_j(x)$ to the matrices $\frac{\partial^2 c_j(x)}{\partial x^2}$, defined by

3.17 $$H_j(x) = \frac{\partial^2 \ell(x,y(x))}{\partial x^2} + \frac{\partial g(x)}{\partial x}^T \frac{\partial y(x)}{\partial x} + \frac{\partial y(x)}{\partial x}^T \frac{\partial g(x)}{\partial x} + \gamma_j \frac{\partial g(x)}{\partial x}^T \frac{\partial g(x)}{\partial x}$$

where $\ell(x,y(x)) = f^0(x) + \langle y(x), g(x) \rangle$. It is shown in [8] that $H_j(x) = \frac{\partial^2 c_j(x)}{\partial x^2}$ for all x such that $g(x) = 0$, $j = 0, 1, 2, \ldots$. We define $\Delta = \{x \in \mathbb{R}^n \mid g(x) = 0, \nabla_x \ell(x,y(x)) = 0\}$, $\Delta_j = \{x \in \mathbb{R}^n \mid \nabla c_j(x) = 0\}$; the maps A_j, t_j, will be defined in the algorithm, with $F = \mathbb{R}^n$.

<u>Algorithm</u>: Parameters: $\alpha \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $0 < \varepsilon_0 << 1$, $0 < \varepsilon_1 << 1$, $\gamma \geq 1$, $\{\gamma_j\}_{j=0}^{\infty}$. Initial guess: $z_0 = x_0$.

<u>Step 0</u>: Set $i = 0$, $j = 0$.

<u>Step 1</u>: If $\nabla c_j(x_i) \neq 0$, go to step 3; else go to step 2.

<u>Step 2</u>: If $g(x_i) = 0$, stop ($x_i \in \{x \in \Delta_i \mid t_j(x) \leq 0\} \subset \Delta$); else go to step 9 (since $t_j(x_i) > 0$ by inspection).

Step 3: If $t_j(x_i) \triangleq -\langle \frac{\partial g(x_i)}{\partial x}^T \left(\frac{\partial g(x_i)}{\partial x}\frac{\partial g(x_i)}{\partial x}^T\right)^{-1} g(x_i), \nabla c_j(x_i)\rangle$ $+ \|g(x_i)\|^2 \leq 0$, go to step 4; else go to step 9.

Comment: The test in step 3, roughly, is on the angle between $\nabla c_j(x_i)$ and the Newton direction $v(x_i) \triangleq \frac{\partial g(x_i)}{\partial x}^T \left(\frac{\partial g(x_i)}{\partial x}\frac{\partial g(x_i)}{\partial x}^T\right)^{-1} g(x_i)$, for solving $g(x) = 0$, defined by $v(x_i) = \arg\min\{\|v\|^2 | g(x_i) + \frac{\partial g(x_i)}{\partial x} v = 0\}$.

Step 4: If $|\det H_j(x_i)| \geq \varepsilon_0$, go to step 5; else go to step 6.

Step 5: If

$$\langle \nabla c_j(x_i), H_{c_j}(x_i)^{-1}\nabla c_j(x_i)\rangle \geq$$

3.18 $$\min\{\varepsilon_1, \|\nabla c_j(x_i)\|^{\gamma}\}\|\nabla c_j(x_i)\| \; \|H_j(x_i)^{-1} \nabla c_j(x_i)\|,$$

set $h(x_i) = - H_j(x_i)^{-1}\nabla c_j(x_i)$ and go to step 7; else go to step 6.

Step 6: Set $h(x_i) = - \nabla c_j(x_i)$.

Note: The two alternatives for $h(x_i)$ (step 5 or step y) define two maps A'_j, A''_j which combine to form a single map A_j, in accordance with the rules governing Algorithm Model 2.

Step 7: Compute the smallest nonnegative integer $\ell_i \geq 0$ such that

3.20 $$c_j(x_i + \beta^{\ell_i} h(x_i)) - c_j(x_i) \leq \beta^{\ell_i}\alpha \langle \nabla c_j(x_i), h(x_i)\rangle$$

Step 8: Set $x_{i+1} = x_i + \beta^{\ell_i} h(x_i) \in A_j(x_i)$, set $i = i+1$, and go to step 1.

Step 9: Set $z_{j+1} = x_i$, set $j = j + 1$ and go to step 1. ¤

Another example conforming to Algorithm Model 4 can be found in [7a].

Conclusion

We have presented in this paper two models for stabilizing algorithms with good local convergence properties and are model for stabilizing an algorithm which requires the automatic selection of a parameter. An examination of the original proofs given for the algorithms which we selected as applications, shows that they satisfy the corresponding assumptions stated in the theorems in this paper.

REFERENCES

1) D. P. Bertsekas, "Combined Primal-Dual and Penalty Methods for Constrained Minimization," to appear in SIAM, J. Control.

2) R. P. Brent, "Some Efficient Algorithms for Solving Systems of Nonlinear Equations," SIAM, J. Num. Anal. 10(2) (1973) 327-344.

3) K. Brown and J. Dennis, "On Newton-Like Iteration Functions: General Convergence Theorems and a Specific Algorithm," Numer. Math. 12, 1968, 186-191.

4) C. Broyden, "Quasi-Newton Methods and their Application to Function Minimization," Math. Comp. 21, 1967, 368-381.

5) R. Fletcher, "A Class of Methods for Nonlinear Programming in the Termination and Convergence Properties," Integer and Nonlinear Programming, J. Abadie (ed.) North Holland 1970.

6) M. R. Hestenes, "Multiplier and Gradient Methods," J. O. T. A. 4(5) 1969, 303-320.

7) L. Kantorovich and G. Akilov, "Functional Analysis in Normed Spaces," Pergamon Press, Oxford, 1964.

8) D. Q. Mayne and E. Polak, "Feasible Directions Algorithms for Optimization Problems with Equality and Inequality Constraints," Univ. of Calif., Berkeley, ERL Memo M457, May 1974.

9) H. Mukai and E. Polak, "A Quadratically Convergent Primal Dual Algorithm with Global Convergence Properties," Univ. of Calif., Berkeley, ERL Memo M455, June 21, 1974.

10) E. Polak, "A Modified Secant Method for Unconstrained Minimization," Math. Prog. 6, (3), 1974, 264-280.

11) E. Polak, "Computational Methods in Optimization," Academic Press N. Y. 1971.

12) E. Polak and I. Teodoru, "Newton Derived Methods for Nonlinear Equations and Inequalities," Nonlinear Programming Symp., Univ. of Wisconsin, Madison, Wisconsin, April 15-17 1974.

13) M. J. D. Powell, "A Method for Nonlinear Constraints in Minimization Problems," <u>Optimization</u>, R. Fletcher(ed) Academic Press, N. Y. 1969.

14) B. N. Pshenichnyi, "Newton's Method for the Solution of Systems of Equalities and Inequalities," <u>Math Notes</u>, Acad. Sc. USSR 8, 827-830, 1970.

15) S. M. Robinson, "Extension of Newton's Method to Mixed Systems of Nonlinear Equations and Inequalities," Tech. Sum. Report No. 1161, Math. Research Center, Univ. of Wisconsin, 1971.