PHONE CALL MAKE A COPY

Date May 23/89 Time 10:29

To Sue

WHILE YOU WERE OUT

M Matt Vranich

of Information Specialists

Phone 216-321-7500

Telephoned	☒	Please call	☒
Called to see you	☐	Will call again	☐
Wants to see you	☐	Returned your call	☐

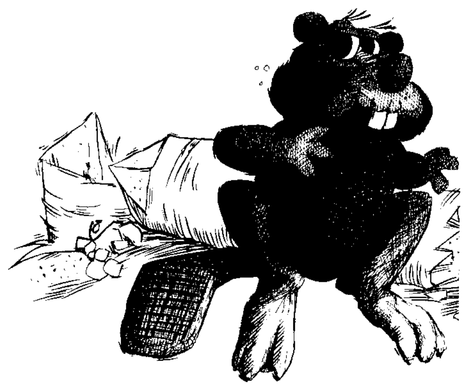
MESSAGE 2490 Lee Blvd.
Cleveland, OHIO
44118
send free of
charge
due to bank charges.

Operator Pat

031020

sent 80-30 & 80-31 today's
May 24

UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



Nonlinear Programming Via an Exact Penalty Function: Asymptotic Analysis

T.F. Coleman
A.R. Conn

CS-80-30

June, 1980

NONLINEAR PROGRAMMING VIA AN EXACT PENALTY FUNCTION: ASYMPTOTIC ANALYSIS

T.F. Coleman

Applied Mathematics Division
Argonne National Laboratory

A.R.Conn

Computer Science
University of Waterloo

ABSTRACT

In this paper we consider the final stage of a 'global' method to solve the nonlinear programming problem [4]. We prove 2-step superlinear convergence. In the process of analyzing this asymptotic behavior, we compare our method (theoretically) to the popular recursive quadratic programming approach.

NONLINEAR PROGRAMMING VIA AN EXACT PENALTY FUNCTION: ASYMPTOTIC ANALYSIS

T.F. Coleman

Applied Mathematics Division
Argonne National Laboratory

A.R.Conn

Computer Science
University of Waterloo

1. Introduction

The nonlinear programming problem can be expressed as

$$\underset{x}{\text{minimize}}\ f(x) \tag{1.1}$$

subject to $\quad \phi_i(x) \geqslant 0, \quad i = 1, \ldots, m,$

where m is a positive integer and f, ϕ_i, $i = 1, \ldots, m$ are continuously differentiable functions mapping $\mathbf{R}^n$ to $\mathbf{R}^1$.

Many algorithms have been proposed to solve (1.1), and recently, recursive quadratic programming has been a popular approach. While this method often exhibits fast local behaviour, it is not a robust global procedure. There have been, and continue to be, attempts to 'globalize' this method (for example, [5], [6], [9], [12], [14]) however to date there does not exist an entirely satisfactory method. We discuss the method of Han [12] in section 3.

In [4], Coleman and Conn introduce a method, based on an exact penalty function, which possesses both global and fast local convergence properties. In [4], numerical results are given which support this claim, and global convergence is proven. It is the intent of this paper to rigorously establish the superlinear properties. In the process we will directly compare our method to the recursive quadratic programming approach.

2. Local Considerations

a) The Algorithm

In this section we will carefully consider the search direction produced by the recursive quadratic programming method when we are 'near' a solution to (1.1). It will be seen that there may be unnecessary computation and storage. A

This work is supported in part by NSERC Grant No. A8639 and the U.S. Dept. of Energy.

geometric interpretation of the search direction leads to a modification which eliminates this excess. This new direction is exactly that produced by the algorithm of Coleman and Conn (derived in [4]) when in a neighbourhood of the solution.

Let x^* be a local solution to (1.1) and suppose that at x^* the active set is $\{1, \ldots, t\}$, where $t \leqslant n$. As in Powell [15], we will assume that we are sufficiently close to x^* so that the active set has been 'identified' and the recursive quadratic programming procedure reduces to the problem

$$\underset{d}{\text{minimize}} \;\; \nabla f(x^k)^T d + \tfrac{1}{2} d^T B_k d \tag{2.1}$$

$$\text{such that} \;\; \phi_i(x^k) + \nabla \phi_i^T d = 0, \quad i = 1, \ldots, t.$$

Using the formulation of Powell [15], the solution to (2.1) can be written as

$$d^k = q^k + r^k,$$

$$\text{where} \;\; q^k = -B_k^{-1} A_k (A_k^T B_k^{-1} A_k)^{-1} \Phi(x^k), \quad \text{and} \tag{2.2}$$

$$r^k = \{B_k^{-1} A_k (A_k^T B_k^{-1} A_k)^{-1} A_k^T B_k^{-1} - B_k^{-1}\} \nabla f(x^k).$$

The $n \times t$ matrix A_k is defined as $(\nabla \phi_1(x^k), \ldots, \nabla \phi_t(x^k))$, and the t-vector $\Phi(x^k)$ is equal to $(\phi_1(x^k), \ldots, \phi_t(x^k))^T$. The $n \times n$ matrix B_k is an approximation to the Hessian of the Lagrangian function.

Provided we start sufficiently close to x^*, a stepsize of unity is assumed in [15], and thus we have

$$x^{k+1} \leftarrow x^k + d^k, \tag{2.3}$$

where d^k is given by (2.2).

It is instructive to introduce the $n \times (n-t)$ matrix Z_k (commonly used by Gill and Murray - see [11], for example) which satisfies

$$A_k^T Z_k = 0, \qquad Z_k^T Z_k = I_{(n-t)}. \tag{2.4}$$

Provided $Z_k^T B_k Z_k$ is positive definite, the solution to (2.1) can be rewritten as

$$d^k = h^k + v^k, \tag{2.5}$$

$$\text{where} \;\; h^k = -Z_k (Z_k^T B_k Z_k)^{-1} Z_k^T (\nabla f(x^k) + B_k v_k), \tag{2.6}$$

$$\text{and} \;\; v^k = -A_k (A_k^T A_k)^{-1} \Phi(x^k). \tag{2.7}$$

We assume that the columns of A_k are linearly independent.

Let $L(x, \lambda)$ denote the Lagrangian function $f(x) - \lambda^T \Phi(x)$, where λ is a t-vector. Suppose that at iteration k we have available x^k and λ^k, estimates to the optimal primal and dual variables, x^* and λ^*. Let B_k be an estimate to the current Lagrangian Hessian,

$$G_f(x^k) - \sum_{i=1}^{t} \lambda_i^k G_{\phi_i}(x^k). \tag{2.8}$$

Equations (2.6) and (2.7) can be interpreted in an interesting geometric fashion. Firstly, v^k is just the least-squares solution to the system

$$\Phi(x^k) + A_k^T v = 0. \tag{2.9a}$$

That is, v^k is a 'Newton-like' attempt to solve the system

$$\Phi(x) = 0, \tag{2.9b}$$

starting at the point x^k, and using exact information computed at x^k. The step h^k can be viewed as an approximation to the constrained Newton step (w.r.t. x) for the Lagrangian (in the manifold spanned by the columns of Z_k and containing the point $x^k + v^k$). This 'Newton' step is based on *approximate* Lagrangian *gradient* information at the point $x^k + v^k$. To see this consider that

$$Z_k^T[\nabla f(x^k) + B_k v^k] = Z_k^T[\nabla f(x^k) - \sum_{i=1}^{t} \lambda_i^k \nabla \phi_i(x^k) + B_k v^k].$$

But B_k approximates $G_f(x^k) - \sum_{i=1}^{t} \lambda_i^k G_{\phi_i}(x^k)$, the Lagrangian Hessian at x^k, which we denote by $G_L(x^k, \lambda^k)$. Thus,

$$h_k = -Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T(\nabla L(x^k, \lambda^k) + B_k v^k), \tag{2.10}$$

which approximates

$$-Z_k(Z_k^T G_L(x^k, \lambda^k) Z_k)^{-1} Z_k^T(\nabla L(x^k, \lambda^k) + G_L(x^k, \lambda^k) v^k). \tag{2.11}$$

But by Taylor's Theorem,

$$Z_k^T[\nabla L(x^k, \lambda^k) + G_L(x^k, \lambda^k) v^k] \approx Z_k^T[\nabla L(x^k + v^k, \lambda^k)]. \tag{2.12}$$

Considering (2.10) and (2.12) it is clear that h^k is an approximation to the constrained Newton direction based on *approximate gradient* information at $x^k + v^k$.

In summary then, the direction d^k can be viewed as a two-part process. First, the step v^k is taken, based on *exact* information at x^k: v^k satisfies $\Phi(x^k + v) = 0$, up to first-order terms. From the point $x^k + v^k$ a step h^k is taken in the space spanned by Z_k: h^k is a 'Newton-like' attempt to satisfy $Z_k^T[\nabla L((x^k + v^k) + h)] = 0$, however only *approximate gradient* information is used at $x^k + v^k$.

It is difficult to imagine improving on the step v^k (up to first-order) since v^k uses exact information. The question should be asked however — is h^k a good approximation to the true constrained Newton direction at $x^k + v^k$? This question is naturally divided into the following questions:

(i) Is $Z_k^T B_k Z_k$ a good approximation (in some sense) to $Z_k^T G_L(x^k, \lambda^k) Z_k$?

(ii) Is $Z_k^T[\nabla L(x^k, \lambda^k) + B_k v^k]$ a good approximation (in some sense) to $Z_k^T[\nabla L(x^k + v^k, \lambda^k)]$?

Interestingly, Powell [15] proved that question (ii) can be ignored, to some extent, (assuming convergence) and yet a 2-step superlinear convergence rate can be maintained. In particular, the accuracy of $Z_k^T B_k v^k$ is not important. This suggests that one could ignore the computation of $Z_k^T B_k v^k$ *altogether*. Specifically, let $d^k = \bar{h} + v^k$, where

$$\bar{h}^k = -Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T \nabla f(x^k). \tag{2.13}$$

We note that since

$$Z_k^T \nabla f(x^k) = Z_k^T(\nabla f(x^k) - \sum_{i=1}^{t} \lambda_i^k \nabla \phi_i(x^k)), \tag{2.14}$$

we can interpret $\bar{h}^k$ as an approximation to the constrained Lagrangian Newton direction, starting at x^k, (in the manifold containing x^k and spanned by the columns of Z_k) based on *exact* gradient information. If we view v^k as being added after $\bar{h}^k$, then v^k is now an attempt to solve $\Phi(x^k + \bar{h}^k + v) = 0$, based on *old* information (that is, A and Φ are computed at x^k, not $x^k + \bar{h}^k$). Nevertheless, it can be shown that the iterate

$$x^{k+1} \leftarrow x^k + \bar{h}^k + v^k,$$

will result in a 2-step superlinear convergence rate.

We note that

(i) $\bar{h}^k + v^k$ is *not* a solution to the quadratic programming problem (2.1),

(ii) Only the *projected* Hessian, $Z_k^T G_L(x^k, \lambda^k) Z_k$, need be computed.

[Murray & Wright [14], suggest algorithms which, at times, also ignore the term $Z_k^T B_k v^k$.]

Since we are now viewing v^k as being taken after $\bar{h}^k$, and since v^k is based on information evaluated at x^k, it seems reasonable to suggest that v^k be 'improved' by re-evaluating gradients and functions at $x^k + \bar{h}^k$. Such computation would probably be unjustifiably expensive; however, global convergence considerations [4] demand that the active constraint functions, ϕ_i, $i = 1, \ldots, t$, be evaluated at $x^k + \bar{h}^k$. (This does not destroy 2-step superlinearity.) Thus we define

$$\bar{v}^k = -A_k(A_k^T A_k)^{-1} \Phi(x^k + \bar{h}^k), \tag{2.15}$$

$$\text{and set} \quad x^{k+1} \leftarrow x^k + \bar{h}^k + \bar{v}^k. \tag{2.16}$$

We emphasize that the only new information that is obtained at $x^k + \bar{h}^k$ is the vector function value $\Phi(x^k + \bar{h}^k)$. The matrix A_k is not re-computed at $x^k + \bar{h}^k$, but contains gradient information accurate at x^k. (Thus, matrix decompositions are *not* modified.) We note that properties i) and ii) above continue to hold for the step $\bar{h}^k + \bar{v}^k$.

Based on the preceding observations, we present the following 'local' algorithm. This local method is exactly that to which the global procedure of Coleman and Conn [4] automatically reduces to in a neighbourhood of a solution.

Algorithm 1 (Local)

(0) Select an x^0 *sufficiently close* to x^* and set $k \leftarrow 1$.

(1) Determine the dual estimates $\{\lambda^k\}$.

(2) 'Update' $Z_k^T B_k Z_k$ maintaining positive definiteness.

(3) Determine h^k:

$$\text{Solve} \qquad (Z_k^T B_k Z_k)\tilde{h} = -Z_k^T \nabla f(x^k),$$
$$\text{and set} \quad \bar{h}^k \leftarrow Z_k \tilde{h}.$$

(4) Determine $\bar{v}^k$:

$$\bar{v}^k \leftarrow -A_k(A_k^T A_k)^{-1}\Phi(x^k+\bar{h}^k).$$

(5) Update:

$$x^{k+1} \leftarrow x^k + \bar{h}^k + \bar{v}^k,$$
$$\text{go to (1).}$$

Note:

(i) This algorithm statement is not meant to reflect the actual implementation. This question is dealt with in [4].

(ii) Theoretically, it does not matter how step (1) is performed as long as $\{\lambda^k\} \rightarrow \lambda^*$, where $\nabla f(x^*) = \sum_{i=1}^{t} \lambda_i^* \nabla \phi_i(x^*)$. In practise we use the least-squares solution to

$$A_k \lambda = \nabla f(x^k),$$

computed using a QR decomposition of A_k (see [4]).

Next we establish that Algorithm 1 generates a sequence $\{x^k\}$, which (under a convergence assumption) satisfies

$$\frac{\|x^{k+1} - x^*\|}{\|x^{k-1} - x^*\|} \rightarrow 0.$$

(b) 2-step Superlinear Convergence

Before stating and proving the major result of this section, a number of preliminary results are established. We make the following assumptions:

A) f, ϕ_i, $i = 1, \ldots, m$ are twice continuously differentiable,

B) the second-order sufficiency conditions (as in Fiacco and McCormick [10]) are satisfied at x^*,

C) $\{x^k\}$ is generated by Algorithm 1, and $\{x^k\} \in W$, a compact set,

D) the columns of $A(x) = (\nabla\phi_1(x), \ldots, \nabla\phi_t(x))$ are linearly independent for all $x \in W$.

We first establish that the horizontal step, $\bar{h}^k$, is bounded by the distance between x^k and x^*.

Lemma 1

Under assumptions A) - D) and assuming that there exist scalars b_1, b_2 $(0 < b_1 \leqslant b_2)$ such that

$$b_1\|y\|^2 \leqslant y^T(Z_k^T B_k Z_k)y \leqslant b_2\|y\|^2, \quad \forall k, \forall y \in \mathbf{R}^{n-t} \tag{2.17}$$

then there exists an $L_1 > 0$ such that

$$\|\bar{h}^k\| \leqslant L_1\|x^k - x^*\|.$$

Proof

By Algorithm 1,

$$\begin{aligned} h^k &= -Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T \nabla f(x^k) \\ &= -Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T(\nabla L(x^k, \lambda^*)). \end{aligned}$$

But, by (2.17), $\{(Z_k^T B_k Z_k)^{-1}\}$ is bounded above, thus there exists an $\tilde{L}_1 > 0$ such that

$$\|\bar{h}^k\| \leqslant \tilde{L}_1\|\nabla L(x^k, \lambda^*)\|. \tag{2.18}$$

But $\nabla L(x^*, \lambda^*) = 0$, and thus using Lipschitz continuity the result follows. □

(Note: unless stated otherwise, $\|\cdot\|$ denotes the 2-norm.)

A similar bound exists for $\bar{v}^k$.

Lemma 2

Under assumptions A) - D), there exists an L_2 such that

$$\|\bar{v}^k\| \leqslant L_2\|x^k - x^*\|.$$

Proof

From Algorithm 1,

$$\bar{v}^k = -A_k(A_k^T A_k)^{-1}\Phi(x^k + \bar{h}^k).$$

But $\quad \phi_j(x^k + \bar{h}^k) = \phi_j(x^k) + O(\|\bar{h}^k\|^2).$

Thus, $\|\bar{v}^k\| \leqslant \|A_k\| \cdot \|(A_k^T A_k)^{-1}\| \cdot \|\Phi(x^k)\| + O(\|\bar{h}^k\|^2).$

But $\Phi(x^*) = 0$. Thus using Lipschitz continuity of ϕ_i, and the boundedness of A_k, $(A_k^T A_k)^{-1}$, the result follows. □

Lemma 3

Under the assumptions of Lemmas 1 and 2, there exists an $L_3 > 0$ such that

$$\|x^{k+1} - x^*\| \leqslant L_3\|x^k - x^*\|.$$

Proof

Follows directly from Lemmas 1 and 2. □

Clearly, by definition, the columns of (A_k, Z_k) span $\mathbf{R}^n$. Therefore, we can write

$$x^{k+1} - x^* = A_k w^{k+1} + Z_k u^{k+1}, \tag{2.19}$$

$$\text{and} \quad x^k - x^* = A_k w^k + Z_k u^k, \tag{2.20}$$

where w^k, $w^{k+1} \in \mathbf{R}^t$, u^k, $u^{k+1} \in \mathbf{R}^{n-t}$.
Using these definitions, the following lemma is easy to establish.

Lemma 4

Under the stated assumptions,

$$\text{if} \quad \frac{\| w^{k+1} \|}{\| x^k - x^* \|} \to 0, \quad \text{and} \quad \frac{\| u^{k+1} \|}{\| x^{k-1} - x^* \|} \to 0,$$

$$\text{then} \quad \frac{\| x^{k+1} - x^* \|}{\| x^{k-1} - x^* \|} \to 0.$$

Proof: Obvious. □

Lemma 4 suggests that 2-step superlinear convergence can be proved in a separable fashion: we show separately that $\dfrac{\| w^{k+1} \|}{\| x^k - x^* \|} \to 0$, and $\dfrac{\| u^{k+1} \|}{\| x^{k-1} - x^* \|} \to 0.$

Theorem 1

Under the assumptions of Lemmas 1 and 2, and assuming that $\{x^k\} \to x^*$, then

$$\frac{\| w^{k+1} \|}{\| x^k - x^* \|} \to 0.$$

Proof

From Algorithm 1,

$$x^{k+1} = x^k - A_k(A_k^T A_k)^{-1} \Phi(x^k + \bar{h}^k) + \bar{h}^k, \tag{2.21}$$

$$= x^k - A_k(A_k^T A_k)^{-1} \Phi(x^k) + \bar{h}^k + y^k, \tag{2.22}$$

where $\| y^k \| = O(\| \bar{h}^k \|^2)$. But for each $j \in \{1, \ldots, t\}$,

$$\phi_j(x^k) = \nabla \phi_j(\xi_j^k)^T (x^k - x^*), \quad \text{where}$$

$$\xi_j^k = x^k + \theta_j^k (x^* - x^k), \quad 0 \leqslant \theta_j^k \leqslant 1.$$

Thus if we define matrices $\tilde{A}_k = (\nabla \phi_j(\xi_j^k), \ldots, \nabla \phi_t(\xi_t^k))$, and $E_k = A_k(A_k^T A_k)^{-1}[\tilde{A}_k^T - A_k^T]$, then (2.22) becomes

$$x^{k+1} = x^k - A_k(A_k^T A_k)^{-1} A_k^T (x^k - x^*) - E_k(x^k - x^*) + \bar{h}^k + y^k. \tag{2.23}$$

Using (2.23) and then multiplying by A_k^T, we obtain

$$A_k^T(x^{k+1} - x^*) = -A_k^T E_k(x^k - x^*) + A_k^T y^k. \tag{2.24}$$

Combining (2.19) with (2.24) gives

$$w^{k+1} = -(A_k^T A_k)^{-1} A_k^T E_k(x^k - x^*) + (A_k^T A_k)^{-1} A_k^T y^k. \tag{2.25}$$

But $\| y^k\| = O(\| \overline{h}^k\|^2)$, and therefore using Lemma 1, assumption D) and compactness, there exists an $L_4 > 0$ such that

$$\| (A_k^T A_k)^{-1} A_k^T y^k\| \leqslant L_4\| x^k - x^*\|^2, \text{ and thus}$$

$$\| w^{k+1}\| \leqslant \| (A_k^T A_k)^{-1}\| \cdot \| A_k^T\| \cdot \| E_k\| \cdot \| x^k - x^*\| + L_4\| x^k - x^*\|^2.$$

But, by definition of $\| E_k\|$ and the convergence assumption, $\{\| E_k\|\} \rightarrow 0$, and therefore our result follows. □

Theorem 2

Under the assumptions of Theorem 1, and assuming that

$$Z_k^T B_k Z_k \rightarrow Z_*^T G_L(x^*, \lambda^*) Z_*, \text{ then}$$

$$\frac{\| u^{k+1}\|}{\| x^{k-1} - x^*\|} \rightarrow 0.$$

(Note: Z_* satisfies $Z_*^T Z_* = I_{(n-t)}$, $A_*^T Z_* = 0$, where $A_* = (\nabla \phi_1(x^*), \ldots, \nabla \phi_t(x^*))$.

Proof

By Algorithm 1,

$$x^{k+1} = x^k - Z_k(Z_k^T B_k Z_k)^{-1} Z_k^T(\nabla L(x^k, \lambda^*)) + \overline{v}^k. \tag{2.26}$$

Define

$$\tilde{E}_k = Z_k[(Z_k^T B_k Z_k)^{-1} - (Z_k^T G_L(x^k, \lambda^*) Z_k)^{-1}] Z_k^T,$$

and combining this with (2.26), we obtain

$$x^{k+1} = x^k - Z_k(Z_k^T G_L(x^k, \lambda^*) Z_k)^{-1} Z_k^T(\nabla L(x^k, \lambda^*) - \tilde{E}_k(\nabla L(x^k, \lambda^*)) + \overline{v}^k. \tag{2.27}$$

Using a Taylor's expansion, there exists a matrix $\tilde{G}_L(x^k, \lambda^*)$ which satisfies

$$\nabla L(x^k, \lambda^*) = \tilde{G}_L(x^k, \lambda^*)(x^k - x^*), \text{ and}$$

$$\tilde{G}_L(x^k, \lambda^*) \rightarrow G_L(x^*, \lambda^*).$$

Let us define a matrix $\overline{E}_k$:

$$\overline{E}_k = Z_k(Z_k^T G_L(x^k, \lambda^*) Z_k)^{-1} Z_k^T[\tilde{G}_L(x^k, \lambda^*) - G_L(x^k, \lambda^*)]. \tag{2.28}$$

In light of (2.28), equation (2.27) can be written as

$$x^{k+1} = x^k - Z_k(Z_k^T G_L(x^k, \lambda^*) Z_k)^{-1} Z_k^T G_L(x^k, \lambda^*)(x^k - x^*) - \overline{E}_k(x^k - x^*) - \tilde{E}_k \tilde{G}_L(x^k, \lambda^*)(x^k - x^*) + \overline{v}^k. \tag{2.29}$$

Let us define $C_k = Z_k(Z_k^T G_L(x^k, \lambda^*) Z_k)^{-1} Z_k^T G_L(x^k, \lambda^*) A_k$. Using this definition and combining (2.20) and (2.29) we obtain

$$x^{k+1} = x^k - Z_k u^k - C_k w^k - \overline{E}_k(x^k - x^*) \qquad (2.30)$$
$$- \tilde{E}_k \tilde{G}_L(x^k, \lambda^*)(x^k - x^*) + \overline{v}^k.$$

Again, if we apply (2.19) and (2.20) and multiply by Z_k^T, then (2.30) reduces to

$$u^{k+1} = -Z_k^T C_k w^k - Z_k^T(\overline{E}_k + \tilde{E}_k \tilde{G}_L(x^k, \lambda^*))(x^k - x^*), \qquad (2.31)$$

$$\Rightarrow \quad \frac{\| u^{k+1}\|}{\| x^{k-1} - x^*\|} \leqslant \| C_k\| \frac{\| w^k\|}{\| x^{k-1} - x^*\|}$$
$$+ (\| \overline{E}_k\| + \| \tilde{E}_k\| \cdot \| \tilde{G}_L\|)\frac{\| x^k - x^*\|}{\| x^{k-1} - x^*\|}.$$

But $\{\| C_k\|\}$ is bounded, and using the convergence assumption $\{\| \overline{E}_k\|\} \to 0$. In addition, by assumption $Z_k^T B_k Z_k \to Z_*^T G_L(x^*, \lambda^*) Z_*$, and thus it follows that $\{\| \tilde{E}_k\|\} \to 0$. But, by Lemma 3, $\left\{\frac{\| x^k - x^*\|}{\| x^{k-1} - x^*\|}\right\}$ is bounded, and by Theorem 1 $\left\{\frac{\| w^k\|}{\| x^{k-1} - x^*\|}\right\} \to 0$, and therefore $\frac{\| u^{k+1}\|}{\| x^{k-1} - x^*\|} \to 0$.

Theorem 3

Under the assumptions

i) $\{x^k\} \to x^*$

ii) There exist scalars b_1, b_2 such that $0 < b_1 \leqslant b_2$, and
$b_1\| y\|^2 \leqslant y^T(Z_k^T B_k Z_k)y \leqslant b_2\| y\|^2$, $\forall k$, $\forall y \in \mathbf{R}^{n-t}$

iii) f, ϕ_i, $i = 1, \ldots, m$ are twice continuously differentiable

iv) second-order sufficiency conditions hold at x^*

v) $\{x^k\}$ is generated by Algorithm 1

vi) $Z_k^T B_k Z_k \to Z_*^T G_L(x^*, \lambda^*) Z_*$

vii) the columns of $A(x^*)$ are linearly independent

then

$$\frac{\| x^{k+1} - x^*\|}{\| x^{k-1} - x^*\|} \to 0.$$

Proof: Follows directly from Lemma 4, and Theorems 1 and 2. □

(c) Local Convergence

Next we establish that Algorithm 1 is locally convergent. That is, provided x^k is sufficiently close to x^*, then $\{x^k\} \to x^*$.

Lemma 5

Let x^0 be some vector in $\mathbf{R}^n$, and suppose that x^1 and x^2 are generated by Algorithm 1, with starting vector x^0. Under assumptions A) – D), if x^0 is sufficiently close to x^*, it follows that

$$\| A_1 w^2\| \leqslant \frac{1}{4}\| x^0 - x^*\|,$$

where $x^2 - x^* = A_1 w^2 + Z_1 u^2$, and A_1, Z_1 are defined in (2.19).

Proof:

By (2.25),

$$\| A_1 w^2 \| \leqslant \| A_1 \|^2 \cdot \| (A_1^T A_1)^{-1} \| \cdot \| E_1 \| \cdot \| x^1 - x^* \| + \| A_1 \|^2 \cdot \| (A_1^T A_1)^{-1} \| \cdot \| y^1 \| . \tag{2.32}$$

Using assumptions A) – D), it is straightforward to establish:

$$\| E_1 \| \leqslant \hat{\beta} \cdot \| x^1 - x^* \| , \tag{2.33}$$

and $$\| y^1 \| \leqslant \bar{\beta} \cdot \| x^1 - x^* \|^2 , \tag{2.34}$$

for some $\hat{\beta} > 0$, $\bar{\beta} > 0$. Using (2.32), (2.33), (2.34), and Lemma 3, there exists $\tilde{\beta} > 0$ such that

$$\| A_1 w^2 \| \leqslant (\tilde{\beta} \| x^0 - x^* \|) \| x^0 - x^* \| .$$

Hence, for $\| x^0 - x^* \| \leqslant \dfrac{1}{(\tilde{\beta} \cdot 4)}$,

$$\| A_1 w^2 \| \leqslant \frac{1}{4} \| x^0 - x^* \| . \quad \square$$

Lemma 6

Under assumptions A) – D), and assuming that

$$Z_k^T B_k Z_k \to Z_*^T G_L(x^*, \lambda^*) Z^* \quad \text{as} \quad x^k \to x^*,$$

then for x^0, $Z_0^T B_0 Z_0$ sufficiently close to x^*, $Z_*^T G_L(x^*, \lambda^*) Z_*$ respectively,

$$\| u^2 \| \leqslant \frac{1}{4} \| x^0 - x^* \| .$$

Proof

By (2.31),

$$\| u^2 \| \leqslant \| C_1 \| \; \| w^1 \| \; + \| \bar{E}_1 + \tilde{E}_1 \tilde{G}_L(x^1, \lambda^*) \| \; \| x^1 - x^* \| . \tag{2.35}$$

Using Lemma 3 and the definitions of $\bar{E}_1$, $\tilde{E}_1$, the result follows analagously to the proof of Lemma 5. $\square$

Theorem 4

Under the asumptions of Lemmas 5 and 6, then for x^0, $Z_0^T B_0 Z_0$ sufficiently close to x^*, $Z_*^T G_L(x^*, \lambda^*) Z_*$ respectively,

$$\{x^k\} \to x^*,$$

where $\{x^k\}$ is generated by Algorithm 1.

Proof

By Lemmas 5,6, and (2.19), $\| x^2 - x^* \| \leqslant \frac{1}{2} \| x^0 - x^* \|$. It follows that $\{x^{2k}\} \to x^*$. But, by Lemma 3, $\{x^{2k+1}\} \to x^*$, and therefore $\{x^k\} \to x^*$. $\square$

3. Global Considerations

Algorithm 1 is, of course, purely local: convergence is proven if the initial estimate, x^0, is sufficiently close to x^*. A 'global' algorithm of Coleman and Conn [4] has the significant property that the method automatically simplifies to Algorithm 1 in a neighbourhood of the solution. Thus a 2-step superlinear rate is also achieved. (We define *global convergence* precisely in [4]: here it is sufficient to say that under 'weak' assumptions we converge to a *local* minimum of the nonlinear programming problem.)

Global convergence is exhibited due to the fact that an exact penalty function is required to decrease (sufficiently) at *each step*. Superlinearity is achieved because a step of $\bar{h}^k + \bar{v}^k$ (as given in Algorithm 1) is guaranteed to decrease the penalty function (sufficiently) in a neighbourhood of x^*: a step of $\bar{h}^k + \bar{v}^k$ is *always* taken for large enough k.

We contrast this with the algorithm of Han [12], [13]: Han proves global convergence for an algorithm based on a recursive programming approach with a superimposed exact penalty used with the line search. Global and superlinear convergence do *not* mesh together however — superlinearity is achieved only in the case where the stepsize is *one*; as we demonstrate in the Concluding Remarks section, a stepsize of one will not guarantee a decrease in the penalty function (a condition required for globality). Thus the full algorithm must switch from a slow global method to a fast superlinear procedure with no assurance of convergence.

4. Concluding Remarks

a) Globality for RQP Directions?

A major consequence of Theorem 3 and the global results given in [4], is that the method of [4] possesses both global and superlinear convergence properties simultaneously. Possessing both properties is due essentially to the fact that a stepsize of unity (which gives superlinearity) will result in a decrease in p, provided we are sufficiently close to x^*, where $p(x) = f(x) - \frac{1}{\mu}\sum_{i=1}^{m} \min(0, \phi_i(x))$. [See, for example, references [1], [2], [3], [7], and [16].] Can recursive quadratic programming also satisfy these two properties? That is, does there exist a neighbourhood of x^* in which a move $x \rightarrow x + d$ will result in a decrease in p, where d is the solution to the quadratic programming problem? As we demonstrate below, the answer (even in the convex case) is, in general, no.

For any x let $d(x)$ be the solution to problem (2.1). (We assume that x is sufficiently close to x^* so that the active set is 'identified'.) Let us make the simplifying assumptions that

i) exact Hessian information is used

ii) the functions ϕ_i, $i = 1, \ldots, t$ are concave.

Define $\hat{G}_L(x) = G_f(x) - \sum_{i=1}^{t} \lambda_i(x) G_{\phi_i}(x)$, where $\{\lambda_i(x)\}$ are the dual variable estimates. Let x' satisfy $\phi_i(x') = 0$, $i = 1, \ldots, t$. Thus, $d(x') = -Z(Z^T\hat{G}_L Z)^{-1}Z^T\nabla f(x)$, and

$$f(x'+d) - f(x')) = -\nabla f^T Z(Z^T \hat{G}_L Z)^{-1} Z^T \nabla f + \tfrac{1}{2} d^T G_f d + o(\|d\|^2).$$
$$= -d^T \hat{G}_L d + \tfrac{1}{2} d^T G_f d + o(\|d\|^2).$$

Also, $\phi_i(x'+d) = \tfrac{1}{2} d^T G_{\phi_i} d + o(\|d\|^2)$. By assumption ii), $\phi_i(x'+d) < 0$, if x' is sufficiently close to x^*, and therefore

$$-\min(0, \phi_i(x'+d)) + \min(0, \phi_i(x')) = -\tfrac{1}{2} d^T G_{\phi_i} d + o(\|d\|^2).$$

Thus, $p(x'+d) - p(x') = -\tfrac{1}{2} d^T [G_f + \sum_{i=1}^{t} (\frac{1}{\mu} - 2\lambda'_i) G_{\phi_i}] d + o(\|d\|^2)$. But it is certainly possible that $\lambda'_i \in (0, \frac{1}{2\mu})$, for $i = 1, \ldots, t$. In these cases it follows that $\sum_{i=1}^{t} (\frac{1}{\mu} - 2\lambda'_i) G_{\phi_i}$ is negative definite and thus we can construct simple convex examples in which the matrix $G_f + \sum_{i=1}^{t} (\frac{1}{\mu} - 2\lambda'_i) G_{\phi_i}$ is negative definite. It follows that $p(x'+d) - p(x)$ is positive. Therefore, in these examples for all δ sufficiently small, there exists an $x' \in N_\delta(x^*)$ such that $p(x'+d) > p(x')$ and d is the recursive quadratic programming direction.

b) Future Work

The convergence rate results presented in this paper are dependent on the projected Hessian approximation asymptotically approaching the true projected Hessian. The full $n \times n$ Lagrangian Hessian is never approximated, and thus computational expense is reduced. To ensure that the projected Hessian approximation approach the true projected Hessian necessitates that an expensive method be used (such as gradient differencing along the columns of Z_k), at least in a neighbourhood of x^*. In fact, the numerical results given in [4] are based on an implementation which uses a rank-2 updating procedure when far from the solution and then switches to a gradient difference method when nearing a solution.

It is expected that a full Quasi-Newton implementation of Algorithm 2 will be developed. This expectation is fueled by the result of Powell [15] which states that, (using the recursive quadratic programming approach), the projected Hessian approximations need only be asymptotically accurate along the directions of search, and superlinearity will be maintained. (This result parallels a superlinearity characterization given by Dennis and Moré [8].) We expect a similar property holds for the method given here and this gives hope for a full Quasi-Newton implementation. □

ACKNOWLEDGEMENTS

The splendid typesetting was accomplished by Anne Trip-de-Roche and Brian Finch using the Photon typesetter and Troff.

REFERENCES

[1] C. Charalambous, *A lower bound for the controlling parameter of the exact penalty function*, Math. Programming 15 278-290 (1978)

[2] C. Charalambous, *On the conditions for optimality of the nonlinear l_1 problem*, Math. Programming 17 pp. 123-135, (1979)

[3] T.F.Coleman and A.R.Conn, *Second-order conditions for an exact penalty function and applications*, to appear in Mathematical Programming, 1980.

[4] T.F.Coleman and A.R. Conn, *Nonlinear programming via an exact penalty function method: Global analysis*. Comp. Sc. Tech. Rep. CS-80-31, University of Waterloo, Waterloo, Ont., 1980.

[5] R.M. Chamberlain, *Some examples of cycling in variable metric methods for constrained optimization*, Math. Programming 16 pp. 378-383 (1979)

[6] R.M. Chamberlain, H.C. Pederson and M.J.D. Powell, *A Technique for Forcing Convergence in Variable Metric Methods for Constrained Optimization*, presented at Tenth International Symposium on Mathematical Programming, Montreal, 1979.

[7] A.R. Conn and T. Pietrzykowski, *A penalty function method converging directly to a constrained optimum*, SIAM J. Numer. Anal. 14 (2), pp. 348-375, (1977).

[8] J.E. Dennis and J.J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev. pp. 46-84 (1977).

[9] L.C.W. Dixon, *On the convergence properties of variable metric recursive quadratic programming methods*, presented at the Tenth International Symposium on Mathematical Programming, Montreal, 1979.

[10] A.V. Fiacco and G.P. McCormick, *Non-linear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.

[11] P.E. Gill and W. Murray, *Numerical Methods for Constrained Optimization*, Academic Press, London, 1974.

[12] S.P. Han, *A globally convergent method for nonlinear programming*, JOTA 22 (3), 1977.

[13] S.P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming 11 pp. 263-282 (1976).

[14] W. Murray and M. Wright, *Projected Lagrangian Methods Based on the Trajectories of Penalty and Barrier Functions*, SOL 78-23, Stanford, Nov. 1978.

[15] M.J.D. Powell, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in *Nonlinear Programming 3* edited by O. Mangasarian, R.Meyer, S. Robinson, Academic Press, pp. 27-63 (1978).

[16] W.I.Zangwill, *Nonlinear programming via penalty functions*, Man. Science 13, pp. 344-350 (1967).