On the Convergance of Constrained Optimization Methods with Accurate Hessian Information on a Subspace

Richard H. Byrd*

CU-CS-270-84 July 1984

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* This research was supported by National Science Foundation Grant #MCS-8115475.

ABSTRACT

In this paper we analyze methods of the type proposed by Coleman and Conn for nonlinearly constrained optimization. We show that if the reduced Hessian approximation is sufficiently accurate, then the method generates a sequence of iterates which converges one-step superlinearly. This result applies to a quasi-Newton implementation. If the reduced exact Hessian is used the method has an R-order equal to that of the secant method. We also prove a similar result for a modified version of successive quadratic programming. Finally some parallels between convergence results for methods that approximate the reduced Hessian, and multiplier methods that use the reduced Hessian inverse are pointed out.

1. Introduction

In this paper we study the local convergence behavior of several algorithms for nonlinearly constrained optimization, and show that their convergence rates are somewhat better than has been thought. Although many methods have been proposed for numerical solution of optimization problems with nonlinear constraints, when we restrict ourselves to equality constraints, and to the operation of the methods near the solution, the variety is not nearly so great. Indeed Tapia [1978] has shown that a number of known methods are equivalent to Newton's method on the Kuhn-Tucker equations with an approximate Hessian. This common method we will refer to as successive quadratic programming (SQP). Consider a problem of the form

$$\underset{x \in R^n}{\text{minimize}} \quad f(x) \tag{1.1}$$

$$\text{subject to} \quad c(x) = 0.$$

where f is a real-valued function on R^n and c maps R^n to R^t, where $t < n$. An iteration of SQP on this problem has the following form:

Algorithm 1.

Given x_k let d_k minimize

$$\nabla f(x_k)^T d + \tfrac{1}{2} d^T B_k d$$

$$\text{subject to} \quad A_k^T d = -c(x_k). \tag{1.2}$$

Let

$$x_{k+1} = x_k + d_k.$$

Here B_k is an $n \times n$ matrix approximating the Hessian of the Lagrangian,

$$\nabla^2 L(x_k, \lambda_k) = \nabla^2 f(x_k) + \Sigma \lambda_k^i \nabla^2 c^i(x_k),$$

with approximations, λ_k^i, to the Lagrange multipliers of problem (1.2). The matrix A_k equals $A(x_k)$ where, for any x, $A(x)$ is an $n \times t$ matrix whose columns are the gradients $\nabla c^i(x)$. The step d_k may be expressed as the solution to the linear system,

$$\begin{bmatrix} Z_k^T B_k Z_k & Z_k^T B_k Y_k \\ 0 & A_k^T Y_k \end{bmatrix} \begin{bmatrix} Z_k^T d \\ Y_k^T d \end{bmatrix} = -\begin{bmatrix} Z_k^T \nabla f(x_k) \\ c(x_k) \end{bmatrix} \tag{1.3}$$

Here Y_k is a matrix of orthogonal columns spanning the column space of A_k, and Z_k is a matrix of orthogonal columns spanning the null space of A_k^T. Of course Y_k and Z_k are not uniquely determined by A_k, but they may be easily computed from a QR factorization of A_k. Note that the projection matrix

$$P_k = Z_k Z_k^T = I - A_k (A_k^T A_k)^{-1} A_k^T$$

only depends on A_k.

In this form it becomes clear that, as shown by Boggs, Tolle, and Wang [1982], the step is independent of the range space reduced Hessian $Y_k^T B_k Y_k$ or equivalently, of $(I-P_k)B_k(I-P_k)$. Thus the convergence behavior of the method will depend only on the matrices $Z_k^T B_k Z_k$ and $Z_k^T B_k Y_k$ or, equivalently, on $P_k B_k$. Indeed, it has been shown by Boggs, Tolle, and Wang [1982] and, under weaker assumptions, by Fontecilla, Steihaug, and Tapia [1983] that the sequence x_k generated by SQP is Q-superlinearly convergent if and only if the sequence

$$\frac{\| P_k (B_k - \nabla^2 L(x_k, \lambda_k))(x_{k+1} - x_k) \|}{\| (x_{k+1} - x_k) \|} \tag{1.4}$$

converges to zero. Throughout this paper when we use the term superlinear convergence without the qualifier Q- or R- we will be referring to Q-superlinear convergence.

We now consider the horizontal-vertical algorithm of Coleman and Conn. We give here only the local version of the method which represents its expected behavior near the solution. Although it is very similar to successive quadratic programming, the two are not equivalent. A more detailed description and motivation may be found in the paper by Coleman and Conn [1982a]. A single iteration is as follows.

Algorithm 2.

Given x_k, let

$$h_k = -Z_k M_k^{-1} Z_k^T \nabla f(x_k)$$

$$v_k = -A_k (A_k^T A_k)^{-1} c(x_k + h_k)$$

$$x_{k+1} = x_k + h_k + v_k .$$

Here M_k is an approximation to $Z_k^T \nabla^2 L(x_k, \lambda_k) Z_k$. Such an approximation could be obtained by using the exact Hessian, by finite difference approximations along the directions given by the columns of Z_k, or by some quasi-Newton method. Note that h_k is a solution of the homogeneous equality constrained quadratic program

$$\text{minimize } \nabla f(x_k)^T h + \tfrac{1}{2} h^T Z_k M_k Z_k^T h$$

$$\text{subject to } A_k^T h = 0.$$

Note also that the constraints are evaluated at two points, x_k and $x_k + h_k$.

If we compare the step generated by this algorithm with the step produced by SQP we see two differences. One is the extra evaluation of the constraints in the Coleman- Conn algorithm; the other is in the Hessian approximation used. In SQP any positive definite nxn matrix may be used. In the Coleman-Conn method only the reduced Hessian is approximated. Effectively the entire Hessian is approximated by $Z_k M_k Z_k^T$, a matrix of rank $n-t$. Thus if were not for the extra constraint evaluation, Coleman and Conn's algorithm would be a special case of SQP with a Hessian approximation satisfying $Z_k^T B_k Y_k = 0$. A similar strategy has been proposed by Womersley and Fletcher [1982], but with $v_k = -A_k (A_k^T A_k)^{-1} c(x_k)$. Because of this difference the main results of Section 2 do not apply to his method.

When we look at the speed of convergence of these two methods we see strong similarities. Since the condition (1.4) clearly cannot be satisfied in general by the Hessian approximation used in effect by Coleman and Conn, $Z_k M_k Z_k^T$, one would not expect superlinear convergence for this method. However Powell [1978] shows that, under the assumption of convergence, if the somewhat weaker condition

$$\frac{||P_k(B_k-\nabla^2 L(x_k,\lambda_k))P_k(x_{k+1}-x_k)||}{||(x_{k+1}-x_k)||} \to 0 \tag{1.5}$$

holds then the sequence is two-step superlinearly convergent. That is

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

This condition is reasonable to ask of the projected Hessian and, since it seems unlikely that an extra evaluation of the constraints will hurt performance, it is not surprising that Coleman and Conn [1982a] were able to prove two-step convergence of their method where the projection of the exact Hessian is used. Later they weakened this assumption on the Hessian approximation and were able to analyze a quasi-Newton type method [Coleman and Conn 1982b].

Recently, examples have been worked out showing that for methods of this type the iterates x_k are not in general one-step superlinearly convergent [Byrd 1984, Yuan 1984]. In spite of these examples, computational experiments indicate that these methods appear to be one step superlinearly convergent. It is also not obvious that being two-step superlinear as opposed to one-step superlinear implies that these methods are therefore slower.

In this paper we will show that under reasonable assumptions, the sequence $\{x_k+h_k\}$ produced by the Coleman-Conn algorithm is actually one step superlinearly convergent, and for exact Hessians we give a convergence order. Analogous results will be proved for a modified version of SQP subject to the condition (1.5). Finally we will point out some analogous two-step and one-step phenomena that occur in multiplier methods.

2. Convergence of the Coleman-Conn Horizontal-Vertical Method

We now show that under the standard conditions, the sequence of points $\{x_k+h_k\}$ produced by the Coleman-Conn algorithm is superlinearly convergent. All theoretical results will be proven subject to the following assumptions.

Assumptions

1. The point x_* is a local minimum of problem (1.1).
2. The functions f and c are twice continuously differentiable in a neighborhood of x_*.
3. The point x_* is a regular point of the constraints, i.e. $A(x_*)$ has full rank. (With assumption 2 this implies that in some neighborhood of x_* the matrices $A(x)$ and $[A(x)^T A(x)]^{-1}$ are bounded uniformly.)
4. The matrix $\nabla^2 L(x_*,\lambda_*)$ is positive definite on the null space of the constraint gradients.

In the following two lemmas we show that both the horizontal and vertical steps are proportional to the current error.

Lemma 2.1. If x_k and x_k+h_k are sufficiently close to x_* then

$$||v_k|| = O(||x_k+h_k-x_*||) \tag{2.1}$$

and

$$||x_{k+1}-x_*|| = O(||x_k+h_k-x_*||). \tag{2.2}$$

Proof By definition of the algorithm

$$v_k = -A_k(A_k^T A_k)^{-1} c(x_k+h_k).$$

By Assumption 2 and Taylor's Theorem,

$$||c(x_k+h_k)|| = O(||x_k+h_k-x_*||),$$

and (2.1) follows from Assumption 3. Since $x_{k+1} = x_k + h_k + v_k$, (2.2) follows also. ■

Lemma 2.2 If x_k and $x_k + h_k$ are sufficiently close to x_* and the quantity

$$\theta_k = \frac{\| [M_k - Z_k{}^T \nabla^2 L(x_*, \lambda_*) Z_k]\, Z_k{}^T (x_{k+1} - x_k) \|}{\| x_{k+1} - x_k \|}$$

is sufficiently small then

$$\| h_k \| = O(\| x_k - x_* \|)$$

and

$$\| x_k + h_k - x_* \| = O(\| x_k - x_* \|).$$

Proof

$$Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k Z_k{}^T h_k = [Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k - M_k]\, Z_k{}^T h_k + M_k Z_k{}^T h_k$$

so

$$\| h_k \| = \| Z_k{}^T h_k \| \le \| [Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k]^{-1} \| \, (\theta_k \| x_{k+1} - x_k \| + \| M_k Z_k{}^T h_k \|)$$

and by construction of h_k

$$\| h_k \| \le \| Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k^{-1} \| \, [\theta_k (\| h_k \| + \| v_k \|) + \| Z_k{}^T \nabla f(x_k) \|]$$

By Lemma 2.1 we know that for some constant γ and for x_k sufficiently close to ||xx||

$$\| v_k \| \le \gamma \| x_k + h_k - x_* \| \le \gamma \| x_k - x_* \| + \gamma \| h_k \|.$$

Thus, if θ_k is sufficiently small, we may rearrange to get

$$\| h_k \| \le \frac{\| [Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k]^{-1} \| [\theta_k \gamma \| x_k - x_* \| + \| Z_k{}^T \nabla f(x_k) \|]}{1 - \| [Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k]^{-1} \| \, \theta_k (1+\gamma)}$$

Note that the vector $Z_k{}^T \nabla f(x_k)$ is clearly $O(\| x_k - x_* \|)$ so that

$$\| h_k \| = O(\| x_k - x_* \|).$$

The second result follows immediately. ■

Now we show that the vertical step results in a superlinear decrease in the vertical component of the error.

Lemma 2.3. If the iterates x_{k-1} and $x_{k-1}+h_{k-1}$ are sufficiently close to x_* and assumptions 2 and 3 hold then,

$$A_k^T(x_k+h_k-x_*) = A_k^T(x_k-x_*) = O(\,||x_{k-1}+h_{k-1}-x_*||\ \ ||x_{k-1}-x_*||).$$

Proof. By definition of h_k it follows that

$$A_k^T(x_k+h_k-x_*) = A_k^T(x_k-x_*),$$

and, by definition of the algorithm,

$$A_k^T(x_k-x_*)=A_{k-1}^T(x_{k-1}+h_{k-1}+v_{k-1}-x^*)-(A_{k-1}^T-A_k^T)(x_k-x_*)$$
$$=-c\,(x_{k-1}+h_{k-1})+c\,(x_*)+A_{k-1}^T(x_{k-1}+h_{k-1}-x_*)-(A_{k-1}^T-A_k^T)(x_k-x_*).$$

Using Taylor's Theorem with assumption 2 and noting that $||A_{k-1}-A(x_{k-1}+h_{k-1})|| = O(\,||h_{k-1}||)$ we see that

$$||A_k^T(x_k-x_*)|| = O([\,||x_{k-1}+h_{k-1}-x_*||+||h_{k-1}||]\ ||x_{k-1}+h_{k-1}-x_*||)$$
$$+O(\,||x_{k-1}-x_k||\ ||x_k-x_*||).$$

But, using Lemmas 2.1 and 2.2,

$$||A_k^T(x_k-x_*)|| = O(\,||x_{k-1}+h_{k-1}-x_*||\ \ ||x_{k-1}-x_*||).$$

■

Now we can prove the essential result, that if M_k is accurate along the horizontal step near the solution, then the step from $x_{k-1}+h_{k-1}$ to x_k+h_k reduces the error significantly.

Lemma 2.4. Suppose that the sequence $\{x_k\}$ is generated by the Coleman Conn algorithm and that Assumptions 1-4 hold. If the point x_{k-1} is sufficiently close to x_* and the quantity

$$\theta_k = \frac{||(M_k-Z_k{}^T\nabla^2L(x_*,\lambda_*)Z_k)Z_k{}^T(x_{k+1}-x_k)||}{||x_{k+1}-x_k||}$$

is sufficiently small then

$$\frac{||x_k+h_k-x_*||}{||x_{k-1}+h_{k-1}-x_*||} = O(\,\theta_k+||x_{k-1}-x_*||). \tag{2.3}$$

Proof. First note that

$$x_k+h_k-x_*=(Y_k Y_k^T+Z_k Z_k^T)(x_k+h_k-x_*),$$

and by Lemma 2.3

$$Y_k Y_k^T(x_k+h_k-x_*)=A_k(A_k^TA_k)^{-1}A_k^T(x_k+h_k-x_*)$$

$$= O(\,||x_{k-1}+h_{k-1}-x_*||\;\;||x_{k-1}-x_*||\,). \tag{2.4}$$

Now we consider the tangent component of the error.

$$Z_k^T\nabla^2L(x_*,\lambda_*)Z_k Z_k^T(x_k+h_k-x_*) = Z_k^T\nabla^2L(x_*,\lambda_*)Z_k Z_k^T h_k +$$

$$Z_k^T\nabla^2L(x_*,\lambda_*)Z_k Z_k^T(x_k-x_*)$$

and since $Z_k Z_k^T+Y_k Y_k^T=I$,

$$Z_k^T\nabla^2L(x_*,\lambda_*)Z_k Z_k^T(x_k+h_k-x_*) = (Z_k^T\nabla^2L(x_*,\lambda_*)Z_k-M_k)Z_k^T h_k \tag{2.5}$$

$$+M_k Z_k^T h_k+Z_k^T\nabla^2L(x_*,\lambda_*)[(x_k-x_*)-Y_k Y_k^T(x_k-x_*)].$$

Consider the four terms on the right separately. By definition of θ_k and using Lemmas 2.1 and 2.2,

$$||(Z_k^T\nabla^2L(x_*,\lambda_*)Z_k-M_k)Z_k^T h_k|| = \theta_k\,||x_{k+1}-x_k|| \tag{2.6}$$

$$= O(\,\theta_k\,||x_k-x_*||\,).$$

The step h_k is constructed to satisfy

$$M_k Z_k^T h_k = -Z_k^T\nabla f(x_k). \tag{2.7}$$

By Taylor's theorem

$$Z_k^T\nabla^2L(x_*,\lambda_*)(x_k-x_*) = Z_k^T(\nabla L(x_k,\lambda_*)-\nabla L(x_*,\lambda_*))+O(\,||x_k-x_*||^2)$$

$$= Z_k^T\nabla f(x_k)+O(\,||x_k-x_*||^2). \tag{2.8}$$

Finally, note that by Lemma 2.3

$$Y_k Y_k^T(x_k-x_*) = O(\,||x_{k-1}-x_*||\;\;||x_{k-1}+h_{k-1}-x_*||\,) \tag{2.9}$$

Now substituting (2.6)-(2.9) into (2.5) we have

$$Z_k^T\nabla^2L(x_*,\lambda_*)Z_k Z_k^T(x_k+h_k-x_*) = O(\,\theta_k\,||x_k-x_*||\,)+O(\,||x_k-x_*||^2)$$

$$+O(\,||x_{k-1}-x_*||\;\;||x_{k-1}+h_{k-1}-x_*||\,).$$

Using Lemma 2.1 this implies

$$Z_k^T(x_k+h_k-x_*) = O(\,\theta_k\,||x_k-x_*||\,)+O(\,||x_{k-1}-x_*||\;\;||x_{k-1}+h_{k-1}-x_*||\,).$$

Together with equation (2.4) this establishes the result. ■

Using this we can immediately prove superlinear convergence of the Coleman- Conn algorithm as long as M_k is consistent along the direction $Z_k{}^T h_k$.

Theorem 2.5. Suppose Algorithm 2 is used and the Assumptions 1-4 as well as the condition

$$\theta_k = \frac{||(M_k - Z_k{}^T \nabla^2 L(x_*, \lambda_*) Z_k) Z_k{}^T (x_{k+1} - x_k)||}{||x_{k+1} - x_k||} \to 0 \qquad (2.10)$$

are satisfied. Then if some iterate x_m is sufficiently close to x_* for m sufficiently large it follows that for the rest of the sequence,

$$\frac{||x_k + h_k - x_*||}{||x_{k-1} + h_{k-1} - x_*||} \to 0.$$

Proof. Immediate consequence of equation (2.3). ■

Note that condition (2.10) is a direct analog of Powell's condition (1.5).

We now want to apply our results to some specific methods for getting M_k. The simplest case is a Newton type method where an exact reduced Hessian is used.

Theorem 2.6. Suppose that Algorithm 2 is used with $M_k = Z_k^T \nabla^2 L(x_k, \lambda_k) Z_k$ and $\lambda_k = -(A_k^T A_k)^{-1} A_k^T \nabla f(x_k)$, and that assumptions 1-4 hold. If x_0 is sufficiently close to x_* then the sequence $\{x_k + h_k\}$ converges one-step superlinearly to x_*. In addition both sequences, $\{x_k + h_k\}$ and $\{x_k\}$, converge to x_* with an R-order of at least $\frac{1}{2}(\sqrt{5}+1)$.

Proof. Since we are using exact Hessian information, by continuity of the derivatives of f and c, $\theta_k = O(||x_k - x_*||)$. Substituting this into equation (2.3) yields

$$||x_k+h_k-x_*|| = O(||x_{k-1}+h_{k-1}-x_*|| \, [||x_k-x_*||+||x_{k-1}-x_*||]) \quad (2.11)$$

and superlinear convergence of $\{x_k+h_k\}$ follows immediately. By Lemma 2.1

$$||x_k+h_k-x_*|| = O(||x_{k-1}+h_{k-1}-x_*|| \; ||x_{k-2}+h_{k-2}-x_*||).$$

This inequality is analogous to the well known inequality provable for the secant method, and, as is shown by Ortega and Rheinboldt [1970, section 9.2], it implies the R-order of the sequence x_k+h_k is at least $\frac{1}{2}(\sqrt{5}+1)$, or approximately 1.618. Since, by Lemma 2.1, $||x_{k+1}-x_*|| = O(||x_k+h_k-x_*||)$ the sequence x_k has R-order equal to at least that of x_k+h_k. ■

Of course this is less than the one-step quadratic convergence one would expect from a complete exact Hessian, but it is a bit more than the R-order of $\sqrt{2}$ for the sequence $\{x_k\}$ which follows from the analysis of Coleman and Conn. The difficulty is that the constraint derivatives used to compute the vertical step from x_k+h_k are evaluated at x_k. If the constraint derivatives were also computed at x_k+h_k and used to compute the vertical step it is easy to see that the $||x_{k-1}-x_*||$ term in equation (2.11) would be replaced by $||x_{k-1}+h_{k-1}-x_*||$ and convergence of the sequence x_k+h_k would be quadratic. Of course this is a fairly expensive modification, and it is not clear that the resultant improvement would be worth the expense in gradient evaluations.

We next consider using a secant update to approximate M_k. Coleman and Conn have shown that if a Davidon-Fletcher-Powell update is used for M_k, then the sequence $\{x_k\}$ is two-step superlinearly convergent. Here we show that the same algorithm results in one-step superlinear convergence for the sequence $\{x_k+h_k\}$.

Theorem 2.7. Suppose Algorithm 2 is used with M_k given by

$$M_{k+1} = M_k + \frac{(y_k - M_k s_k)y_k^T + y_k(y_k - M_k s_k)^T}{s_k^T y_k} - \frac{s_k^T(y_k - M_k s_k)y_k y_k^T}{(s_k^T y_k)^2}$$

where

$$s_k = Z_k{}^T h_k,$$

$$y_k = Z_k{}^T[\nabla L(x_k + h_k, \lambda_k) - \nabla L(x_k, \lambda_k)]$$

and

$$\lambda_k = -(A_k^T A_k)^{-1} A_k^T \nabla f(x_k).$$

Assume that Assumptions 1-4 hold and that Z_k is given by a continuous function $Z(x)$ evaluated at $x = x_k$. If $||x_0 - x_*||$ and $||M_0 - Z(x_*)^T \nabla^2 L(x_*, \lambda_*) Z(x_*)||$ are sufficiently small, then the sequences x_k and $x_k + h_k$ converge to x_* and

$$\frac{||x_k + h_k - x_*||}{||x_{k-1} + h_{k-1} - x_*||} \to 0.$$

Proof. This algorithm is the same one described by Coleman and Conn [1982b]. In Theorem 3.14 of that paper they prove that under the above conditions $x_k \to x_*$. Convergence of $x_k + h_k$ follows from our Lemma 2.2. In the same paper, Theorem 3.5 with Theorem 3.14 shows that under these conditions

$$\theta_k = \frac{||(M_k - Z(x_*)^T \nabla^2 L(x_*, \lambda_*) Z(x_*)) Z_k{}^T (x_{k+1} - x_k)||}{||x_{k+1} - x_k||} \to 0$$

which implies

$$\theta_k = \frac{||(M_k - Z_k{}^T \nabla^2 L(x_*, \lambda_*) Z_k) Z_k{}^T (x_{k+1} - x_k)||}{||x_{k+1} - x_k||} \to 0.$$

The result follows from Theorem 2.5 of this paper. ■

By similar arguments the same result can be proven for the Broyden-Fletcher-Goldfarb-Shanno update.

Note that this algorithm has the drawback that the gradient of the Lagrangian must be computed twice at each step. The theorems of Coleman and Conn show that one superlinear step is made for each four gradient evaluations. Theorem 2.7 shows that one superlinear step is made for each two gradient evaluations, a fact which makes the algorithm somewhat more competitive.

3. A Modified Successive Quadratic Programming Algorithm.

We would like to see if, by analogy with the previous section, some version of successive quadratic programming might be one-step superlinearly convergent whenever it satisfies the Powell condition (1.5). However, the extra evaluation of the constraints at $x_k + h_k$ was essential to proving the 1-step convergence results of section 2, so it seems unlikely that we could prove an analogous result for any standard version of SQP.

What we will do is to consider a modification of SQP that has a similar one-step superlinear convergence when condition (1.5) is satisfied. The modification involves taking an extra vertical step after the standard SQP step is computed. It is described precisely below.

Algorithm 3.

1. Compute $\nabla f(x_k)$, A_k, and B_k.
2. Compute d_k, the solution to

$$\underset{d \in R^n}{\text{minimize}} \; \nabla f(x_k)^T d + \tfrac{1}{2} d^T B_k d \tag{3.1}$$

$$\text{subject to} \; A_k^T d = -c(x_k)$$

3. Compute

$$v_k = -A_k (A_k^T A_k)^{-1} c(x_k + d_k)$$

4. Set

$$x_{k+1} = x_k + d_k + v_k$$

Note that this algorithm differs from the Coleman-Conn method in that an approximation to the entire Hessian is computed and, instead of a horizontal step, a "diagonal" step is computed which satisfies the inhomogeneous constraints. This method is slightly more expensive than SQP in that the constraints must be evaluated at one extra point, x_k+d_k. However this is a small cost, and in addition to giving a better convergence rate, the extra constraint evaluation may provide a better interface with the global strategy. In particular, if a nondifferentiable exact penalty function is used as a merit function with a line search, the vertical step should make a "Maratos effect" much less likely to occur, as this seems not to be a problem with the Coleman-Conn method.

We will now show that the sequence x_k+d_k produced by Algorithm 3 is superlinearly convergent if the matrices B_k satisfy Powell's condition (1.5). Note that to a great extent we are using the same arguments as in section 2 with d_k replacing h_k. For that reason we will abbreviate the proofs when one follows the same pattern as a previous proof.

Lemma 3.1. If x_k and x_k+d_k are sufficiently close to x_* then

$$||v_k|| = O(||x_k+d_k-x_*||)$$

and

$$||x_{k+1}-x_*|| = O(||x_k+d_k-x_*||).$$

Proof. Same as proof of Lemma 2.1 with d_k in place of h_k. ■

Lemma 3.2. If x_k and x_k+d_k are sufficiently close to x_*, the quantity

$$\varphi_k = \frac{||Z_k Z_k^T(B_k-\nabla^2 L(x_*,\lambda_*))Z_k Z_k^T(x_{k+1}-x_k)||}{||x_{k+1}-x_k||}$$

is sufficiently small, and the matrix norms $||B_k||$ are uniformly bounded, then

$$||d_k|| = O(||x_k-x_*||)$$

and

$$||x_k+d_k-x_*|| = O(||x_k-x_*||)$$

Proof The solution to (3.1) may be expressed

$$d_k = -A_k(A_k^T A_k)c(x_k) - Z_k(Z_k^T B_k Z_k)^{-1} Z_k{}^T [B_k A_k (A_k^T A_k)^{-1} c(x_k) + \nabla f(x_k)].$$

This is clear from considering equation (1.3). The first two terms on the right hand side are proportional to $c(x_k)$, and since $||B_k||$ is uniformly bounded they are clearly $O(||x_k - x_*||)$. The last term is equal to h_k. Since the condition $\varphi_k \to 0$ has the same effect as the condition $\theta_k \to 0$ in this context, the results follow from the same argument as was used in the proof of Lemma 2.2 for h_k. ■

Lemma 3.3. If the iterates x_k and $x_k + d_k$ are sufficiently close to x_* and Assumptions 2 and 3 hold then,

$$A_k^T(x_k - x_*) = O(||x_k - x_*||^2). \tag{3.2}$$

and

$$A_k^T(x_k + d_k - x_*) = O(||x_{k-1} + d_{k-1} - x_*|| \; ||x_{k-1} - x_*||). \tag{3.3}$$

Proof. By definition of the algorithm,

$$A_k^T(x_k - x_*) = A_{k-1}^T(x_{k-1} + d_{k-1} + v_{k-1} - x^*) - (A_{k-1}^T - A_k^T)(x_k - x_*)$$

$$= -c(x_{k-1} + d_{k-1}) + c(x_*) + A_{k-1}^T(x_{k-1} + d_{k-1} - x_*) - (A_{k-1}^T - akt)(x_k - x_*).$$

By Taylor's Theorem, continuity of derivatives, and Lemmas 3.1 and 3.2, it follows just as in the proof of Lemma 2.3 that

$$||A_k^T(x_k - x^*)|| = O(||x_{k-1} + d_{k-1} - x_*|| \; ||x_k - x_*||).$$

which is equation (3.2). To derive (3.3) note that

$$A_k^T(x_k + d_k - x_*) = A_k^T(x_k - x_*) - c(x_k)$$

$$= O(||x_k - x_*||^2).$$

■

Lemma 3.4. Suppose that the sequence $\{x_k\}$ is generated by Algorithm 3, that Assumptions 1-4 hold, and that the matrix norms $||B_k||$ are uniformly bounded. If the point x_{k-1} is sufficiently close to x_* and the quantity

$$\varphi_k = \frac{||Z_k^T(B_k - \nabla^2 L(x_*,\lambda_*))Z_k Z_k^T(x_{k+1}-x_k)||}{||x_{k+1}-x_k||}$$

is sufficiently small then

$$\frac{||x_k+d_k-x_*||}{||x_{k-1}+d_{k-1}-x_*||} = O(\varphi_k + ||x_{k-1}-x_*||). \tag{3.4}$$

Proof. Note that

$$x_k+d_k-x_* = (Y_k Y_k^T + Z_k Z_k^T)(x_k+d_k-x_*),$$

and by Lemma 3.3

$$\begin{aligned} Y_k Y_k^T(x_k+d_k-x_*) &= A_k(A_k^T A_k)^{-1}A_k^T(x_k+d_k-x_*) \\ &= O(||x_k-x_*||) \\ &= O(||x_{k-1}+d_{k-1}-x_*|| \ ||x_{k-1}-x_*||). \end{aligned} \tag{3.5}$$

Now consider the tangent component of the error.

$$\begin{aligned} Z_k^T\nabla^2 L(x_*,\lambda_*)Z_k Z_k^T(x_k+d_k-x_*) &= Z_k^T\nabla^2 L(x_*,\lambda_*)Z_k Z_k^T(x_k-x_*) \\ &+ Z_k^T B_k Z_k Z_k^T d_k + Z_k^T(\nabla^2 L(x_*,\lambda_*)-B_k)Z_k Z_k^T d_k \end{aligned}$$

and since $Z_k Z_k^T + Y_k Y_k^T = I$,

$$\begin{aligned} Z_k^T\nabla^2 L(x_*,\lambda_*)Z_k Z_k^T(x_k+d_k-x_*) &= Z_k^T\nabla^2 L(x_*,\lambda_*)(x_k-x_*) + Z_k^T B_k d_k \\ &- Z_k^T\nabla^2 L(x_*,\lambda_*)Y_k Y_k^T(x_k+d_k-x_*) \\ &+ Z_k^T(\nabla^2 L(x_*,\lambda_*)-B_k)Z_k Z_k^T d_k. \end{aligned} \tag{3.6}$$

Consider the four terms on the right separately. By Taylor's theorem

$$\begin{aligned} Z_k^T\nabla^2 L(x_*,\lambda_*)(x_k-x_*) &= Z_k^T(\nabla L(x_k,\lambda_*)-\nabla L(x_*,\lambda_*)) + O(||x_k-x_*||^2) \\ &= Z_k^T\nabla f(x_k) + O(||x_k-x_*||^2), \end{aligned} \tag{3.7}$$

and the step d_k is constructed to satisfy

$$Z_k^T B_k d_k = -Z_k^T\nabla f(x_k). \tag{3.8}$$

By Lemma 3.3

$$Z_k^T\nabla^2 L(x_*,\lambda_*)Z_k Y_k Y_k^T(x_k-x_*) = O(||x_{k-1}-x_*|| \ ||x_{k-1}+d_{k-1}-x_*||). \tag{3.9}$$

By definition of φ_k and using Lemmas 3.1 and 3.2,

$$||Z_k^T(\nabla^2 L(x_*,\lambda_*)-B_k)Z_k Z_k^T d_k|| = ||Z_k^T(\nabla^2 L(x_*,\lambda_*)-B_k)Z_k Z_k^T(x_{k+1}-x_k)||$$

$$= \varphi_k \, \|x_{k+1} - x_k\|$$

$$= O(\varphi_k \, \|x_k - x_*\|). \tag{3.10}$$

Now substituting (3.7)-(3.10) into (3.6) we have

$$Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k Z_k^T (x_k + d_k - x_*) = O(\|x_k - x_*\|^2)$$

$$+ O(\theta_k \|x_k - x_*\|) + O(\|x_{k-1} - x_*\| \; \|x_{k-1} + d_{k-1} - x_*\|).$$

By Lemma 3.1 and Assumption 4 this implies

$$Z_k^T (x_k + d_k - x_*) = O(\theta_k \|x_k - x_*\|) + O(\|x_{k-1} - x_*\| \; \|x_{k-1} + h_{k-1} - x_*\|).$$

Together with equation (3.5) this establishes the result. ■

Theorem 3.5. Suppose Algorithm 3 is used, $\|B_k\|$ are uniformly bounded, and Assumptions 1-4 as well as the condition

$$\varphi_k = \frac{\|(M_k - Z_k^T \nabla^2 L(x_*, \lambda_*) Z_k) Z_k^T (x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} \to 0$$

are satisfied. Then if some iterate x_m is sufficiently close to x_* with m large enough it follows that for the rest of the sequence,

$$\frac{\|x_k + d_k - x_*\|}{\|x_{k-1} + d_{k-1} - x_*\|} \to 0.$$

Proof. This is an immediate consequence of equation (3.4). ■

Note that this result involves the same assumption (1.5) as the corresponding result of Powell, but because of the extra constraint evaluation a better convergence rate is achieved.

4. Related phenomena in the method of multipliers.

An interesting parallel to the two step convergence behavior discussed above occurs in certain versions of the diagonalized method of multipliers. In methods such as SQP or the Coleman-Conn method two-step superlinear convergence occurs when the Hessian of the Lagrangian reduced to the null space of the constraint gradients is approximated. The diagonalized method of multipliers may be regarded as dual to these methods in that Lagrange multipliers play a more central role. Below, we indicate that two-step convergence behavior in these methods occurs when only the reduced inverse Hessian on the range space of the constraint gradients is used.

The method of multipliers for solving problem (1.1) involves generating a sequence $\{(x_k, \lambda_k)\}$ where x_k is a solution to the problem

$$\underset{x \in R^n}{\text{minimize}} \; L(x, \lambda_k, \rho) = f(x) + \lambda_k^T c(x) + \tfrac{1}{2}\rho c(x)^T c(x) \tag{4.1}$$

and

$$\lambda_k = U(x_{k-1}, \lambda_{k-1}).$$

If the scalar ρ is chosen sufficiently large, a strong solution (x_*, λ_*) to (1.1) corresponds to an x_* which minimizes $L(x, \lambda_*, \rho)$, and a λ_* which maximizes

$$\underset{x \in R^n}{\text{minimum}} \; L(x, \lambda, \rho).$$

Many multiplier update formulas $U(x, \lambda)$ have been proposed [see Bertsekas, 1982]. A central idea is to choose U to correspond to a method such as Newton's method or steepest descent on the problem of maximizing the minimum value of (4.1) as a function of λ.

As a more efficient alternative Tapia [1977] proposed the diagonalized method of multipliers. Here again one generates a sequence x_k, λ_k with λ_k chosen as above, but with x_k being the result of one Newton or quasi-Newton step on problem (4.1) taken from x_{k-1}. Tapia showed that if the update formula

$$U(x, \lambda) = [\nabla c(x)^T \nabla^2 L(x, \lambda)^{-1} \nabla c(x)]^{-1} \tag{4.2}$$
$$[c(x) - \nabla c(x)^T \nabla^2 L(x, \lambda)^{-1} \nabla f(x)] - \rho c(x)$$

is used and x_k is generated by a Newton step then the sequence $\{x_k, \lambda_k\}$ converges to the Kuhn-Tucker pair (x_*, λ_*) quadratically. Tapia also showed that this procedure is mathematically equivalent to SQP using the exact Hessian of $L(x, \lambda, \rho)$ with respect to x.

A related update formula is the one due to Buys[1972]

$$U(x,\lambda) = \lambda+[\nabla c(x)^T \nabla^2 L(x,\lambda)^{-1} \nabla c(x)]^{-1} c(x) \tag{4.3}$$

If x_k is chosen to solve (4.1) exactly then updates (4.2) and (4.3) coincide.

Note that the Buys formula only requires the inverse of the Hessian of the Lagrangian reduced to the range space of the constraints, while the formula due to Tapia requires the entire Hessian inverse. Byrd [1978] showed that if the Buys update is used in the diagonalized method of multipliers and if x_k is computed from a Newton step, then the sequence $\{(x_k, \lambda_k)\}$ converges to the Kuhn-Tucker pair at a two-step quadratic rate. Thus, just as in the primal SQP type methods, only having Hessian information on a subspace determined by the constraint gradients results in a reduction in convergence rate from one-step quadratic to two-step quadratic.

Byrd [1978] also derived a result parallel to Theorem 3.5. If we modify the diagonalized method of multipliers so that x_k is the result of two successive Newton steps rather than one, and use the Buys update, the pair (x_k, λ_k) still converges only two step quadratically. However, if in this method, we look at the "out of phase" iterates consisting of λ_k and the value of x resulting from the first of the two Newton steps, this pair converges one-step quadratically.

However this correspondence is not perfect; in the diagonalized method of multipliers with either update formula we require the entire Hessian in order to take the Newton step. Another difference between the primal and dual cases is experimentally observed. In computational experiments using the Buys update the observed convergence was clearly two-step and not one-step quadratic in almost all cases. With the Coleman-Conn method or with SQP the observed convergence rate usually appears one-step and it is only in contrived examples [Byrd 1984, Yuan 1984] that two-step behavior is really observed. It is not clear why this is the case.

References

D.P. Bertsekas [1982], *Constrained Optimization and Lagrange Multiplier Methods*, Chapter 2, Academic Press, New York.

P.T. Boggs, J.W. Tolle and P. Wang [1982]. "On the local convergence of quasi-Newton methods for constrained optimization", *SIAM J. Control and Optimization 20*, pp. 161-171.

J.D. Buys [1972], "Dual algorithms for constrained optimization", Ph.D. thesis, Rijksuniversiteit te Leiden, Netherlands.

R.H. Byrd [1978], "Local convergence of the diagonalized method of multipliers", *J. of Optimization Theory and Applications*, *26*, pp. 483-498.

R.H. Byrd [1984], "An example of irregular Convergence in some constrained optimization methods that use the projected Hessian", Technical Report, Dept. of Computer Science, University of Colorado.

T.F. Coleman and A.R. Conn [1980] "Nonlinear programming via an exact penalty function method: global analysis", Report CS-80-31, Computer Science Department, University of Waterloo.

T.F. Coleman and A.R. Conn [1982a] "Nonlinear programming via an exact penalty function method: asymptotic analysis", *Mathematical Programming 24*, pp. 123-136.

T.F. Coleman and A.R. Conn [1982b] "On the local convergence of a quasi-Newton method for a nonlinear programming problem", TR 82-509, Computer Science Department, Cornell University.

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

J.M. Ortega and W.C. Rheinboldt [1970], *Iterative solution of nonlinear equations in several variables*, Academic Press, New York.

M.J.D. Powell [1978]. "The convergence of variable metric methods for nonlinearly constrained optimization calculations", in *Nonlinear Programming 3*, O. Mangasarian, R. Meyer, and S. Robinson, eds., pp.27-63, Academic Press, New York.

R.A. Tapia [1977]. "Diagonalized Multiplier Methods and quasi-Newton methods for constrained optimization", *J. Opt. Theory and Applics. 22*, pp. 135-194.

R.A. Tapia [1978], "Quasi-Newton methods for constrained optimization: equivalence of existing methods and a new implementation", in *Nonlinear Programming 3*, O. Mangasarian, R. Meyer, and S. Robinson, eds., pp.125-164, Academic Press, New York.

R.S. Womersley and R. Fletcher [1982], "An algorithm for composite nonsmooth optimization problems", Dept. of Mathematics Report NA/60, University of Dundee.

Y. Yuan [1984], On the reduced Hessian method for constrained optimization", Technical Report DAMTP 1984/NA1, Dept. of Theoretical Physics and Applied Math, Cambridge University.