NBER WORKING PAPER SERIES

SOLVING SYSTEMS OF NONLINEAR EQUATIONS BY BROYDEN'S METHOD WITH PROJECTED UPDATES

David M. Gay*

Robert B. Schnabel**

Working Paper No. 169

COMPUTER RESEARCH CENTER FOR ECONOMICS AND MANAGEMENT SCIENCE
National Bureau of Economic Research, Inc.
575 Technology Square
Cambridge, Massachusetts 02139

March 1977

Preliminary

*NBER Computer Research Center. Research conducted in part during a visit to the Atomic Energy Research Establishment, Harwell, England, and supported in part by National Science Foundation Grant MCS76-00324 to the National Bureau of Economic Research, Inc.

**Computer Science Dept., Cornell University. Research conducted in part during a visit to the Atomic Energy Research Establishment, Harwell, England, and supported in part by a National Science Foundation Graduate Fellowship.

Abstract

We introduce a modification of Broyden's method for finding a zero of n nonlinear equations in n unknowns when analytic derivatives are not available. The method retains the local Q-superlinear convergence of Broyden's method and has the additional property that if any or all of the equations are linear, it locates a zero of these equations in $n+1$ or fewer iterations. Limited computational experience suggests that our modification often improves upon Broyden's method.

CONTENTS

Section		Page
1.	Introduction	1
2.	The New Method	5
3.	Behavior on Linear or Partly Linear Problems	17
4.	Local Q-Superlinear Convergence on Nonlinear Problems	25
5.	Computational Results	34
6.	Summary and Conclusions	39
7.	References	40

1. Introduction

This paper is concerned with solving the problem

$$\begin{aligned} &\text{given a differentiable} \quad F : \mathbb{R}^n \to \mathbb{R}^n \ , \\ &\text{find} \quad x^* \in \mathbb{R}^n \quad \text{such that} \quad F(x^*) = 0 \end{aligned} \tag{1.1}$$

when derivatives of F are either inconvenient or very costly to compute. We denote the n component functions of F by

$$f_i : \mathbb{R}^n \to \mathbb{R} \quad i = 1,\ldots,n$$

and the Jacobian matrix of F at x by $F'(x)$, $F'(x)_{ij} = \dfrac{\partial f_i}{\partial x_j}(x)$.

When $F'(x)$ is cheaply available, a leading method for the solution of (1.1) is Newton's method, which produces a series of approximations $\{x_1, x_2, \ldots\}$ to x^* by starting from approximation x_o and using the formula

$$x_{i+1} = x_i - F'(x_i)^{-1} F(x_i) \ . \tag{1.2}$$

If F is nonsingular and Lipschitz continous at x^* and x_o is sufficiently close to x^*, then the algorithm converges Q-quadratically to x^* - i.e., there exists a constant c such that $||x_{i+1} - x^*|| \leq c\, ||x_i - x^*||^2$ for all i and some vector norm $||\cdot||$ (c.f. §9.1 of [Ortega & Rheinholdt, 1970]). If F is linear with nonsingular Jacobian matrix, then $x_1 = x^*$.

When $F'(x)$ is not readily available, an obvious strategy is to replace $F'(x_i)$ in (1.2) by an approximation B_i. This leads to the modified Newton iteration

$$x_{i+1} = x_i - B_i^{-1} F(x_i) \tag{1.3a}$$

$$B_{i+1} = U(B_i) \tag{1.3b}$$

where U is some update formula that uses current information about F. Broyden [1965] introduced a family of update formulae U known as quasi-Newton updates. He also proposed the particular update used in "Broyden's method", which we consider in more detail below. If x_o is sufficiently close to x^*, the matrix norm of $B_o - F'(x_o)$ is sufficiently small and several reasonable conditions on F are met, then Broyden's method converges Q-superlinearly to x^*—i.e., $\lim_{i\to\infty} \frac{||x_{i+1} - x^*||}{||x_i - x^*||} = 0$ [Broyden, Dennis & Moré, 1973]. However for linear F, convergence may take as many as $2n$ steps—and $B_{2n} - F'(x^*)$ may have rank $n-1$ (see [Gay, 1977]).

In this paper, we introduce a new method of form (1.3) using an update (1.3b) which is different from but related to Broyden's update. Our new method is still locally Q-superlinearly convergent under the conditions for which Broyden's method is. It has the additional property that if F is linear with nonsingular Jacobian matrix, then $x_i = x^*$ for some $i \leq n+1$, and if $k+1$ iterations are required, then $B_{k+1} - F'(x^*)$ has rank $n-k$. Initial tests show our method to be somewhat superior in performance to Broyden's method.

The basic idea behind our new method is related to one originally proposed by Garcia-Palomares [1973]. Davidon [1975] used this idea independently in deriving a new method for the unconstrained minimization problem,

$$\min_{x \in \mathbb{R}^n} f(x) \quad , \quad f : \mathbb{R}^n \to \mathbb{R} \quad .$$

Davidon also modified an existing update formula to produce a quasi-Newton method which does not use exact line searches but is exact on quadratic problems. This new method has been an improvement in practice. While it has not yet been shown to retain the local superlinear convergence of the method it modified, Schnabel [1977] uses the techniques of this paper to show that a very similar modification retains Q-superlinear convergence as well as the properties of Davidon's [1975] method.

In Section 2 we briefly describe Broyden's method and the important features of quasi-Newton methods. We then introduce our new algorithm in two forms: Algorithm I, a simplified version which is sufficient to discuss its basic and linear properties, and Algorithm II, the version used in practice and to prove local superlinear convergence. We also derive the basic properties of our method which we will use in subsequent sections.

In Section 3 we discuss the behavior of our algorithm on linear problems. We show that if any or all of the equations f_i are linear, then our new algorithm will find a zero of these equations in n+1 or fewer iterations. We also discuss the effect of a certain restart procedure on our algorithm.

In Section 4 we show that our new method is locally Q-superlinearly convergent on a wide class of problems. We discuss our computational results in Section 5 and summarize our results in Section 6.

Henceforth, $||\cdot||$ will denote the ℓ_2 vector norm

$(||v|| = (\sum_{i=1}^{n} v_i^2)^{1/2}$ for $v = (v_1,\ldots,v_n)^T \in \mathbb{R}^n$ or the corresponding

matrix norm, while $||\cdot||_F$ will denote the Frobenius matrix norm:

$$||M||_F = (\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}^2)^{1/2} \quad \text{for } M = (m_{ij}) \in \mathbb{R}^{n\times n} \ .$$

2. The New Method

Quasi-Newton methods are often damped: they take the form

$$x_{i+1} = x_i - \lambda_i B_i^{-1} F(x_i) \tag{2.1a}$$

$$B_{i+1} = U(B_i) \tag{2.1b}$$

where the damping factor $\lambda_i > 0$ is chosen to promote convergence from starting points x_o which may lie outside the region of convergence of the corresponding direct prediction method (1.3). When it leads to a "successful" step, e.g. reduction of $||F||$, the choice $\lambda_i = 1$ is usually preferred.

Broyden's ("good") method is a method of form (2.1), using the update equation

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i)\, s_i^T}{s_i^T s_i} \quad , \quad \text{where} \tag{2.2}$$

$$s_i = \Delta x_i = x_{i+1} - x_i \quad , \tag{2.3a}$$

$$y_i = \Delta F_i = F(x_{i+1}) - F(x_i) \quad . \tag{2.3b}$$

Because of equation (2.2), B_{i+1} satisfies $B_{i+1}\, \Delta x_i = \Delta F_i$. Since for small Δx_i, $F'(x_{i+1})\, \Delta x_i \simeq \Delta F_i$, we expect that B_{i+1} resembles

$F'(x_{i+1})$ in the direction of our last step. Since we have no other information which would help approximate $F'(x_{i+1})$, it is reasonable to change B_i — which hopefully approximates $F'(x_i)$—as little as possible consistent with $B_{i+1}\,\Delta x_i = \Delta F_i$. This suggests the rank one change

$$B_{i+1} = B_i + \frac{(y_i - B_i\, s_i)\, v_i^T}{v_i^T\, s_i} \quad , \tag{2.4}$$

for any vector $v_i \in \mathbb{R}^n$ such that $v_i^T\, s_i \neq 0$. The choice $v_i = s_i$, which yields Broyden's method, minimizes the ℓ_2 or Frobenius norm (the ℓ_2 norm of the elements) of $(B_{i+1} - B_i)$ over all possibilities (2.4) [Dennis and Moré, 1977].

Broyden defined quasi-Newton methods to be those of form (1.3) which satisfy the "quasi-Newton" equation,

$$B_{i+1}\, s_i = y_i \quad , \tag{2.5}$$

in their attempt to build Jacobian approximations. Broyden's method, with intelligent choice of λ_i in (2.1a), has been the most successful quasi-Newton method for solving systems of nonlinear equations.

It is interesting to compare Newton's and Broyden's methods on linear problems where $F(x) = Ax + b$ and A is nonsingular. Whereas Newton's method (1.2) yields $x_i = x^*$ for $i \geq 1$, Broyden's method may require $2n$ direct prediction ($\lambda_i = 1$) steps to produce the exact

solution [Gay, 1977]. In part this is because B_i may never equal A, even though $F'(x) = A$ for all x_i. We can easily see why this may be so. After one iteration we will have $B_1 s_o = y_o$ ($= A s_o$ for a linear problem); after the next iteration we will have $B_2 s_1 = y_1$ ($= A s_1$), but not in general $B_2 s_o = y_o$. At each step we introduce into B_{i+1} our most current information about A; but in doing so we destroy other good information about A learned through previous iterations. Therefore we will never have $B_i = A$, so the iteration $x_{i+1} = x_i - B_i^{-1} F(x_i)$ may take twice as many steps to converge as might seem necessary.

From the preceeding analysis, we are interested in finding an update equation which, while giving $B_{i+1} s_i = y_i$, also retains $B_{i+1} s_j = y_j$ whenever $j < i$ and $B_i s_j = y_j$. Note however that for any formula of form (2.4), $B_{i+1} s_i = y_i$; we can retain old information by our choice of v_i : if $B_i s_j = v_j$ and $v_i^T s_j = 0$, then $B_{i+1} s_j = y_j$. These considerations lead to our new algorithm, given in simplified form as Algorithm I below.

We choose our update at each iteration to be the B_{i+1} which minimizes the Frobenius norm of $B_{i+1} - B_i$ among all B_{i+1} satisfying $B_{i+1} s_i = y_i$ and $(B_{i+1} - B_i) s_j = 0$ for all $j < i$. In Theorem 2.1 we show that the unique solution to this problem is given by update (2.4) with v_i the projection of s_i perpendicular to all the s_j's, $j < i$. The proof is similar to Dennis and Moré's [1977] proof that Broyden's method is the least-change update among all B_{i+1} satisfying $B_{i+1} s_i = y_i$.

Theorem 2.1 Let $B \in \mathbb{R}^{nxn}$ and s, y be non-zero vectors $\in \mathbb{R}^n$ with $Bs \neq y$. Let Z be an m dimensional subspace of $\mathbb{R}^n$, $m < n$. Then for $||\cdot||_N$ either the ℓ_2 or the Frobenius norm, a solution to

$$\min \{||\bar{B} - B||_N \,|\, \bar{B}s = y, \quad (\bar{B} - B)z=0 \text{ for all } z \in Z\} \tag{2.6}$$

is

$$\hat{B} = B + \frac{(y - Bs)\, v^T}{v^T s} ,$$

where v is the orthogonal projection of s onto the orthogonal complement of Z, i.e.,

$$v = s - \sum_{i=1}^{m} \frac{s^T z_i}{z_i^T z_i} z_i$$

with $(z_1,\ldots,z_m)$ an orthogonal basis for Z. The solution is unique in the Frobenius norm.

Proof: Let $S = \{\bar{B} | \bar{B}s = y, (\bar{B} - B)\, z = 0 \ \forall\ z \in Z\}$. Now $\hat{B}s = y$; and since $v^T z_i = 0$ for $i = 1,\ldots, m$, $v^T z = 0$ for all $z \in Z$. Thus $\hat{B} \in S$.

Now consider any $\bar{B} \in S$. Since $y = \bar{B}s$,

$$\hat{B} - B = \frac{(\bar{B} - B)s\, v^T}{v^T s} .$$

Define $d = \sum_{i=1}^{m} \frac{s^T z_i}{z_i^T z_i} z_i = s - v$. Since $d \in Z$ and v is perpendicular to Z, $v^T d = 0$. Thus $v^T s = v^T v$. Since $(\bar{B} - B)\, z = 0$ for all $z \in Z$, $(\bar{B} - B)d = 0$, so $(\bar{B} - B)s = (\bar{B} - B)v$. Therefore $(\hat{B} - B) = \frac{(\bar{B}-B)vv^T}{v^T v}$, so for $||\cdot||_N$ the ℓ_2 or Frobenius norm,

$$||\hat{B} - B||_N \leq ||\bar{B} - B||_N \, ||\frac{vv^T}{v^T v}|| = ||\bar{B} - B||_N \ .$$

Thus $\hat{B}$ is a solution to (2.6). It is the unique solution in the Frobenius norm because the function $\delta: \mathbb{R}^{n\times n} \to \mathbb{R}$ given by $\delta(\bar{B}) = ||\bar{B} - B||_F$ is strictly convex over all $\bar{B}$ in the convex set S. ■

Algorithm I

Let $x_o \in \mathbb{R}^n$, $B_o \in \mathbb{R}^{n\times n}$, $F : \mathbb{R}^n \to \mathbb{R}^n$ be given.

For i=0, 1, 2,...

Choose nonzero $s_i \in \mathbb{R}^n$ (likely $s_i = -\lambda_i B_i^{-1} F(x_i)$)

$$x_{i+1} = x_i + s_i \tag{2.7a}$$

If $F(x_{i+1}) = 0$ then stop

$$y_i = F(x_{i+1}) - F(x_i) \tag{2.7b}$$

$$Q_i = \sum_{j=0}^{i-1} \frac{\hat{s}_j \hat{s}_j^T}{\hat{s}_j^T \hat{s}_j} \tag{2.7c}$$

$$\hat{s}_i = s_i - Q_i s_i \tag{2.7d}$$

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i) \hat{s}_i^T}{\hat{s}_i^T s_i} . \tag{2.7e}$$

Algorithm I is unsuitable for computer implementation for several reasons--most importantly, if $i > n$, then $\hat{s}_i$ will be zero vector. However, it is sufficient for deriving the basic properties of our algorithm (for general functions F) in Theorem 2.2 below; and is also sufficient for discussing the behavior of our algorithm on linear problems in Section 3.

We use the notation $\langle a,b \rangle$ to denote the scalar product

$$a^T b = \sum_{i=1}^{n} a_i b_i \ , \ a, b \in \mathbb{R}^n .$$

Theorem 2.2 Given $x_o \in \mathbb{R}^n$, $B_o \in \mathbb{R}^{nxn}$, $F : \mathbb{R}^n \to \mathbb{R}^n$, let the sequences $\{s_o,\ldots,s_i\}$, $\{y_o,\ldots,y_i\}$, $\{B_o,\ldots,B_{i+1}\}$ be generated by Algorithm I. Define $\hat{s}_o,\ldots,\hat{s}_i$ as in Algorithm I and let $\hat{y}_j = y_j - B_j Q_j s_j$, $j = 0,\ldots,i$. Then at each iteration i, if $s_o,\ldots,s_i$ are linearly independent, then B_{i+1} is well defined and

$$\langle \hat{s}_i, \hat{s}_k \rangle = 0 \qquad k = 0, \ldots, i-1 \tag{2.8a}$$

$$\langle \hat{s}_i, s_k \rangle = 0 \qquad k = 0, \ldots, i-1 \tag{2.8b}$$

$$\langle \hat{s}_i, s_i \rangle = \langle \hat{s}_i, \hat{s}_i \rangle \tag{2.8c}$$

$$B_{i+1} s_k = y_k \qquad k = 0, \ldots, i \tag{2.8d}$$

$$B_{i+1} \hat{s}_k = \hat{y}_k \qquad k = 0, \ldots, i \ . \tag{2.8e}$$

Proof It is straightforward to prove (2.8 a-d) by induction. In view of (2.8a) and (2.7e), it suffices to consider $k = i$ in (2.8e). Using (2.7e), (2.8c), and the definition of $\hat{y}_i$, we find

$$\begin{aligned} B_{i+1} \hat{s}_i &= B_i \hat{s}_i + (y_i - B_i s_i) \\ &= B_i \hat{s}_i + (y_i - B_i Q_i s_i) - B_i (s_i - Q_i s_i) \\ &= B_i \hat{s}_i + \hat{y}_i - B_i \hat{s}_i = \hat{y}_i \ , \end{aligned}$$

so that (2.8e) holds for $k = i$. ∎

Theorem 2.2 shows that we are selecting $\hat{s}_i$ in Algorithm I to be orthogonal to all previous steps s_j, $j<i$, so that we do not disturb information contributed by previous quasi-Newton equations. The equations (2.8e) can be thought of at each iteration as the part of the quasi-Newton equation giving information in the subspace where previous iterations gave none.

Note that if B_i and B_{i+1} are nonsingular, then (2.7e) is equivalent to

$$B_{i+1}^{-1} = B_i^{-1} + \frac{(s_i - B_i^{-1} y_i)\, \hat{s}_i^T B_i^{-1}}{\hat{s}_i^T B_i^{-1} y_i} . \tag{2.9}$$

Therefore if B_o is nonsingular and $\langle \hat{s}_j, B_j^{-1} y_j \rangle \neq 0$ for $0 \leq j \leq i$, then B_{i+1}^{-1} exists, i.e., B_{i+1} is nonsingular.

We now state, in general form, the version of our new algorithm which is used in practice and in proving local Q-superlinear convergence. It recognizes that, in general, the projection of s_i orthogonal to the subspace spanned by $s_o, \ldots, s_{i-1}$ must be the zero vector for some $i \leq n$. The algorithm therefore "restarts" by setting $\hat{s}_i = s_i$ if $\hat{s}_i$ is too small compared to s_i (which must happen at least every n steps).

Theorem 2.2 is still valid if we consider only the vectors s_i, $\hat{s}_i$, y_i, $\hat{y}_i$ generated since the last restart. Since the version of Theorem 2.2 applicable to Algorithm II is needed in Section 4, it is stated as Theorem 2.3. The omitted proof is almost identical to that of Theorem 2.2. Because of the restart criteria, s_i is always strongly linearly independent of all s_j's since the last restart.

Algorithm II

Let $x_o \in \mathbb{R}^n$, $B_o \in \mathbb{R}^{n \times n}$, $F : \mathbb{R}^n \to \mathbb{R}^n$, $\varepsilon > 0$, $\tau > 1$ be given.

Set $\ell_{-1} = 0$.

For $i = 0, 1, 2, \ldots$

Choose nonzero $s_i \in \mathbb{R}^n$ (likely $s_i = -\lambda_i B_i^{-1} F(x_i)$)

$$x_{i+1} = x_i + s_i \tag{2.10a}$$

If $||F(x_{i+1})|| < \varepsilon$ then stop

$$y_i = F(x_{i+1}) - F(x_i) \tag{2.10b}$$

$$Q_i = \sum_{j=\ell_{i-1}}^{i-1} \frac{\hat{s}_j \hat{s}_j^T}{\hat{s}_j^T \hat{s}_j} \tag{2.10c}$$

$$\text{If } ||s_i|| \geq \tau \, ||s_i - Q_i s_i|| \tag{2.10d}$$

then ($\hat{s}_i = s_i$ and $\ell_i = i$)

else ($\hat{s}_i = s_i - Q_i s_i$ and $\ell_i = \ell_{i-1}$)

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i) \hat{s}_i^T}{\hat{s}_i^T s_i} \; . \tag{2.10e}$$

<u>Theorem 2.3</u> Given $x_o \in \mathbb{R}^n$, $B_o \in \mathbb{R}^{nxn}$, $F : \mathbb{R}^n \to \mathbb{R}^n$, $\varepsilon > 0$, $\tau > 1$, let the sequences $\{s_o,\ldots,s_i\}$, $\{y_o,\ldots,y_i\}$, $\{B_o,\ldots,B_{i+1}\}$ be generated by Algorithm II. Define $\{\hat{s}_o,\ldots,\hat{s}_i\}$ as in Algorithm II; let $\hat{y}_j = y_j$ if $\hat{s}_j = s_j$ and $\hat{y}_j = y_j - B_j Q_j s_j$ otherwise, $j = 0,\ldots,i$. Then at each iteration i, $s_{\ell_i},\ldots,s_i$ are linearly independent, B_{i+1} is well defined, and

$$\langle \hat{s}_i, \hat{s}_k \rangle = 0 \qquad k = \ell_i,\ldots,i-1 \tag{2.11a}$$

$$\langle \hat{s}_i, s_k \rangle = 0 \qquad k = \ell_i,\ldots,i-1 \tag{2.11b}$$

$$\langle \hat{s}_i, s_i \rangle = \langle \hat{s}_i, \hat{s}_i \rangle \tag{2.11c}$$

$$B_{i+1}\, s_k = y_k \qquad k = \ell_i, \ldots, i \tag{2.11d}$$

$$B_{i+1}\, \hat{s}_k = \hat{y}_k \qquad k = \ell_i, \ldots, i \tag{2.11e}$$

$$||s_i|| < \tau ||\hat{s}_i|| \tag{2.11f}$$

$$i - \ell_i < n \,. \blacksquare \tag{2.11g}$$

We finally note that the entire subject of quasi-Newton methods for nonlinear systems of equations can be approached by directly forming approximations H_i to $F'(x_i)^{-1}$, the inverse of the Jacobian matrix of F at x. In this case we require $H_{i+1}\, y_i = s_i$ and can achieve this through the rank-one update

$$H_{i+1} = H_i + \frac{(s_i - H_i\, y_i)\, w_i^T}{w_i^T\, y_i} \tag{2.12}$$

for any vector $w_i \in \mathbb{R}^n$ such that $w_i^T\, y_i \neq 0$. We have already seen from (2.9) that if B_i is non-singular, Broyden's update simply corresponds to $w_i = B_i^{-T} s_i$ in (2.12).

The choice of w_i in (2.12) which minimizes the Frobenius norm of $(H_{i+1} - H_i)$ is $w_i = y_i$. The quasi-Newton method using this update was also proposed by Broyden and is sometimes called "Broyden's bad method", because it doesn't perform as well as Broyden's method (update (2.4)) in

practice. However, it has also been demonstrated by Broyden, Dennis, and Moré [1973] to have local superlinear convergence under reasonable assumptions on F.

Similarly, we can propose algorithms I' and II', which update approximations H_i to $F'(x_i)^{-1}$, and choose w_i in (2.12) to be the projection of y_i orthogonal to (some of) the previous y_j's. For instance, Algorithm II' would only require replacing (2.10c-e) with

$$Q_i' = \sum_{j=\ell_{i-1}}^{i-1} \frac{\hat{y}_j \hat{y}_j^T}{\hat{y}_j^T \hat{y}_j} \tag{2.13a}$$

$$\text{If } \|y_i\| \geq \tau \|y_i - Q_i' y_i\|$$

$$\text{then } (\hat{y}_i = y_i \text{ and } \ell_i = i) \tag{2.13b}$$

$$\text{else } (\hat{y}_i = y_i - Q_i' y_i \text{ and } \ell_i = \ell_{i-1})$$

$$H_{i+1} = H_i + \frac{(s_i - H_i y_i) \hat{y}_i^T}{\hat{y}_i^T y_i} \quad . \tag{2.13c}$$

Using Algorithms I' or II' we can prove theorems analagous to 2.2 and 2.3; and we can prove the same convergence results for linear and general nonlinear functions F as are proven in Sections 3 and 4. (As a matter of fact, the proofs of Section 3 are then a bit nicer as they never need assume B_i^{-1} non-singular). We have tested both algorithms II and II'

in practice, and have found that Algorithm II appears more likely to converge than II'.

3. Behavior on Linear or Partly Linear Problems

In this section we examine the behavior of our algorithm on systems of n equations in n unknowns, some or all of which are linear. We find that our algorithm will always locate a zero of whichever of the equations are linear in n+1 or fewer iterations. This property is not shared by Broyden's method.

Theorems 3.1 and 3.2 examine the behavior of Algorithm I on a completely linear system. In reality we would not expect to use our algorithm to solve linear equations. However, it is possible that near a solution, a system of nonlinear equations may be almost linear--and these theorems then tell us what sort of behavior to expect.

Theorem 3.1 shows that if Algorithm I is applied to $F(x) = Ax + b$, A nonsingular, then x_i will equal $x^* = -A^{-1}b$ for some $i \leq n+1$; and if $n+1$ iterations are required, then $B_n = A$. Following Powell [1976], however, we are really more interested in Theorem 3.2, which shows what happens if we do a restart while solving a linear system of equations. This is likely to be the case if we enter a linear region after the algorithm starts. Theorem 3.2 shows that we still require at most n+2 iterations to find x^*, but Example 3.3 shows that B_{n+1} may not equal A.

Theorems 3.4 and 3.5 examine the behavior of Algorithm I when some but not necessarily all of the component functions of F are linear. This may be the most important case in section 3, as partly linear systems do arise in practice; they may also approximate the behavior of a nonlinear system near a solution.

Theorem 3.4 shows that our method will locate a zero of the linear components in n+1 or fewer iterations--and if n+1 iterations are required, then B_n will also agree with the Jacobian matrix on the rows corresponding to the linear equations. Theorem 3.5 shows that in this case, subsequent updates by any rank-one formula (2.4) will not disturb the correct linear information and as long as we take quasi-Newton steps of length one ($\lambda_i = 1$ in (2.1a)), we will only visit points at which the linear components are zero.

Theorems 3.1, 3.2, and 3.4 are stated for simplicity for Algorithm I. They are also true for Algorithm II, which we really use, as long as the algorithm doesn't restart prematurely (i.e., $||s_i|| < \tau\, ||s_i - Q_i s_i||$ in (2.10d) when $i - \ell_{i-1} < n$). Since τ is set significantly larger than 1 in practice, we often expect our theorems to hold for Algorithm II. The conclusions of Theorem 3.5 do not depend on which of the two algorithms we are using.

We denote the subspace spanned by vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ by $[v_1, \ldots, v_k]$; and the column space of matrix $M \in \mathbb{R}^{nxn}$ by $C(M)$.

Theorem 3.1 Let $A \in \mathbb{R}^{nxn}$ be non-singular; $b \in \mathbb{R}^n$; and $F(x) = Ax + b$: $\mathbb{R}^n \to \mathbb{R}^n$. Consider Algorithm I acting on F, starting from any $x_o \in \mathbb{R}^n$ and $B_o \in \mathbb{R}^{nxn}$. If $s_o, \ldots, s_{n-1}$ are linearly independent, then $B_n = A$; and if $s_n = -B_n^{-1} F(x_n)$ then $F(x_{n+1}) = 0$. Moreover, if for some $k < n$, $s_o, \ldots, s_{k-1}$ are linearly independent, B_k^{-1} exists and $B_k^{-1} F(x_k) \in [s_o, \ldots, s_{k-1}]$, and if $s_k = -B_k^{-1} F(x_k)$, then $F(x_{k+1}) = 0$.

Proof: If $s_0,\ldots,s_{n-1}$ are linearly independent, then by Theorem 2.2, $B_n s_i = y_i$, $i = 0,\ldots,n-1$. Since $y_i = F(x_{i+1}) - F(x_i) = A\, s_i$, we have $B_n s_i = A\, s_i$, $i = 0,\ldots,n-1$, so that $B_n = A$.

If $s_0,\ldots,s_{k-1}$ are linearly independent, then by the same reasoning as above, $B_i s_i = A\, s_i$, $i = 0,\ldots,k-1$. Thus if

$s_k = -B_k^{-1} F(x_k) \in [s_0,\ldots,s_{k-1}]$, then $B_k\, s_k = A\, s_k$. Therefore

$$F(x_{k+1}) = F(x_k) + A\, s_k = F(x_k) + B_k\, s_k = F(x_k) + B_k\, [-B_k^{-1} F(x_k)] = 0.$$

■

From the proof of Theorem 3.1, we see that if Algorithm II is acting on a linear problem, then after n-m iterations in which $s_0,\ldots,s_{n-m-1}$ are linearly independent and no restarts have occurred, B_{n-m} will agree with A in n-m directions--i.e., $(A - B_{n-m})$ will have rank m. It is possible--especially if we have entered a linear region after we began--that we will then do a restart: set $\hat{s}_{n-m} = s_{n-m}$ and $\ell_{n-m} = n-m$. Following Powell [1976], we wonder if the information from these n-m iterations is of help. In Theorem 3.2 we show that it is: using quasi-Newton steps (1.3a), we require at most m+2 additional iterations, or a total of n+2, to locate the zero of F. Our conclusions are not as general as Powell's for Davidon's [1975] new unconstrained optimization algorithm, as they do not allow for subsequent restarts or completely general steps; however, our conditions should mirror the behavior of Algorithm II in practice. Also, in our case Example 3.3 shows that the full m+2 iterations may be required and that B_{m+1} may still not equal A.

Theorem 3.2 Let $A \in \mathbb{R}^{nxn}$, $b \in \mathbb{R}^n$, and $F(x) = Ax + b$: $\mathbb{R}^n \to \mathbb{R}^n$. Consider Algorithm I started from $x_o \in \mathbb{R}^n$ and $B_o \in \mathbb{R}^{nxn}$ nonsingular with rank $(A - B_o) = m \geq 1$. Suppose s_i is selected by $s_i = -\lambda_i B_i^{-1} F(x_i)$ and if $s_i \notin [s_o,\ldots,s_{i-1}]$, assume B_{i+1} is nonsingular. Then there exists $j \leq m + 1$ such that $s_j \in [s_o,\ldots,s_{j-1}]$; and if $\lambda_j = 1$, then $F(x_{j+1}) = 0$.

Proof: We first show that for any update of form (2.4)--one of which is used by Algorithm I-- that $s_j \in [s_o,\ldots,s_{j-1}]$ for some $j \leq m + 1$. We accomplish this by showing by induction that if $s_o,\ldots,s_{i-1}$ are linearly independent, then

$$s_i \in [s_o, C(I - B_o^{-1}A)] \tag{3.1}$$

$$(s_i - B_i^{-1}y_i) \in C(I - B_o^{-1}A). \tag{3.2}$$

For $i = 0$, (3.1) is trivially true, and
$s_o - B_o^{-1} y_o = s_o - B_o^{-1} A s_o = (I - B_o^{-1}A) s_o \in C(I - B_o^{-1}A)$.

Assume (3.1-2) true for $i = 0,\ldots,k$. Then

$$-\frac{1}{\lambda_{k+1}} \cdot s_{k+1} = B_{k+1}^{-1} F(x_{k+1}) = B_{k+1}^{-1} [F(x_k) + y_k)].$$

By Theorem 2.2, $B_{k+1}^{-1} y_k = s_k$; and using the inverse form of (2.4) we have

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k)\, v_k^T B_k^{-1}}{v_k^T B_k^{-1} y_k}, \quad \text{so}$$

$$B_{k+1}^{-1} F(x_k) = B_k^{-1} F(x_k) + (s_k - B_k^{-1} y_k) \frac{\langle v_k, B_k^{-1} F(x_k)\rangle}{\langle v_k, B_k^{-1} y_k\rangle}.$$

Since $B_k^{-1} F(x_k) = -\frac{1}{\lambda_k} s_k$, we have $s_{k+1} \varepsilon [s_k, (s_k - B_k^{-1} y_k)]$; so by the induction hypothesis (3.1-2) for $i = k$, $s_{k+1} \varepsilon [s_o, C(I - B_o^{-1}A)]$, which shows (3.1) for $i = k+1$. To complete the induction, $s_{k+1} - B_{k+1}^{-1} y_{k+1} =$

$$= s_{k+1} - \left[B_o^{-1} + \sum_{j=0}^{k} \frac{(s_j - B_j^{-1} y_j)\, v_j^T B_j^{-1}}{v_j^T B_j^{-1} y_j} \right] y_{k+1}$$

$$= s_{k+1} - B_o^{-1} y_{k+1} + \sum_{j=0}^{k} (s_j - B_j^{-1} y_j) \frac{\langle v_j, B_j^{-1} y_{k+1} \rangle}{\langle v_j, B_j^{-1} y_j \rangle} .$$

Since $s_{k+1} - B_o^{-1} y_{k+1} = (I - B_o^{-1}A)\, s_{k+1}$ and $(s_j - B_j^{-1} y_j) \varepsilon C\,(I-B_o^{-1}A)$ for $j \leq k$ by the induction hypothesis, we see that (3.2) holds for $i = k+1$.

Because the subspace $[s_o, C\,(I - B_o^{-1}A)]$ has dimension at most $m+1$, we must have $s_j \varepsilon [s_o, \ldots s_{j-1}]$ for some $j \leq m+1$. Now $B_j s_i = y_i$, $i = 0, \ldots j-1$ by Theorem 2.2; and $B_j s_i = A s_i$, $i = 0, \ldots, j-1$ since F is linear. Therefore $B_j s_j = A s_j$, and $F(x_{j+1}) = F(x_j) + A s_j = F(x_j) + B_j [-B_j^{-1} F(x_j)] = 0$. ∎

<u>Example 3.3</u>. Let $F(x) \equiv x : \mathbb{R}^n \to \mathbb{R}^n$ ($\Rightarrow F'(x) = I$). Consider Algorithm I, with $s_i = -B_i^{-1} F(x_i)$ started from $x_o = (1, \ldots, 1, 2)^T$ and

$$B_o = \left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 1 & 1 & \ddots & & & & & \vdots \\ \vdots & \ddots & \ddots & & & & & \vdots \\ \vdots & & \ddots & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & & & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & & 1 & 0 & 1 & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & & 1 & 0 \\ 1 & \cdots & \cdots & 1 & 0 & \cdots & 0 & 1 \end{array}\right]$$

$$\underbrace{\qquad\qquad}_{m} \quad \underbrace{\qquad\qquad}_{n-m}$$

with $1 \leq m < n$. Then rank $(I-B_o) = m$. Algorithm I then requires a full $m+2$ iterations to reach $x^* = 0$, and rank $(I - B_{m+1}) = 1$. The intermediate values are:

$$s_o = (-1,0,\ldots,0,-1)^T, \qquad \hat{s}_o = s_o ,$$

$$s_j = (\underbrace{0,\ldots,0}_{j},-1,0,\ldots,0)^T , \qquad \hat{s}_j = s_j, \qquad j = 1,\ldots,m-1 ,$$

$$x_j = (\underbrace{0,\ldots,0}_{j},1,\ldots,1)^T , \qquad j = 1,\ldots,m ,$$

$$B_j = \begin{bmatrix} 1 & 0_{1x(j-1)} & 0\cdots\cdots 0 & 0_{mx(n-m-1)} & 0 \\ 1/2 & & 1 \ddots & & -1/2 \\ \vdots & I_{j-1} & \ddots \ddots & I_{n-m-1} & \vdots \\ & & \ddots \ddots 0 & & \\ 1/2 & 0_{(n-j)x(j-1)} & \underbrace{1\cdots\cdots 1}_{m-j} & 0_{1x(n-m-1)} & -1/2 \\ & & & & 1/2 \end{bmatrix} , \quad j = 1,\ldots,m ,$$

$$s_m = (0,\underbrace{-1,\ldots,-1}_{m-1},-2,\ldots,-2)^T , \qquad x_{m+1} = (0,-1,\ldots,-1)^T ,$$

$$\hat{s}_m = s_m - (s_o+\ldots+s_{m-1}) = (1,\underbrace{0,\ldots,0}_{m-1},-2,\ldots,-2,-1)^T ,$$

$$B_{m+1} = B_m + \frac{1}{4(n-m)-2}(0,-1,\ldots,-1)^T \hat{s}_m^T .$$

Therefore $s_{m+1} = (0,1,\ldots,1)^T$; $x_{m+2} = 0$; and $(I - B_{m+1}) = s_{m+1}\,(-1/2 - t,0,\ldots,0,2t,\ldots,2t,\ 1/2 + t)$, where $t = -1/[4(n-m)-2]$. ■

We now consider the case when some but not necessarily all of the component functions of F are linear. For ease of notation we assume that the first m component function of F are linear--however the positioning of the linear functions has

no bearing on the algorithm or the proof. The Jacobian of F will therefore be constant in its first m rows, and we will denote our Jacobian approximations B_i by $\left[\begin{array}{c} C_i \\ \hline D_i \end{array}\right]$, $C_i \in \mathbb{R}^{mxn}$, $D_i \in \mathbb{R}^{(n-m)xn}$.

Theorem 3.4 Let $A \in \mathbb{R}^{mxn}$, $1 \leq m \leq n$; $b \in \mathbb{R}^m$;

$$F(x) = \begin{bmatrix} F_1(x) \\ \hline F_2(x) \end{bmatrix}: \mathbb{R}^n \to \mathbb{R}^n \text{ with } F_1(x) = Ax + b: \mathbb{R}^n \to \mathbb{R}^m \text{ and } F_2: \mathbb{R}^n \to \mathbb{R}^{n-m}.$$

Consider Algorithm I acting on F, starting from any $x_o \in \mathbb{R}^n$ and $B_o \in \mathbb{R}^{nxn}$. If for some $k \leq n$, $s_o,\ldots,s_{k-1}$ are linearly independent, B_k^{-1} exists and $B_k^{-1} F(x_k) \in [s_o,\ldots,s_{k-1}]$, then the choice $s_k = -B_k^{-1} F(x_k)$ leads to $F_1(x_{k+1}) = 0$. Furthermore if $s_o,\ldots,s_{n-1}$ are linearly independent, then $C_n = A$.

Proof: Suppose $s_o,\ldots,s_{k-1}$ are linearly independent and B_k^{-1} exists. By Theorem 2.2, $B_k s_i = y_i$, $0 \leq i \leq k-1$. Since the first m components of y_i are $F_1(x_{i+1}) - F_1(x_i) = A s_i$, while the first m components of $B_k s_i$ equal $C_k s_i$, we have $C_k s_i = A s_i$, $0 \leq i \leq k-1$. In particular, if $k = n$ then this implies $C_n = A$. Moreover, if $B_k^{-1} F(x_k) \in [s_o,\ldots,s_{k-1}]$ (which will necessarily hold for some $k \leq n$) and $s_k = -B_k^{-1} F(x_k)$, then this implies $C_k s_k = A s_k$; because $C_k B_k^{-1} = [I_m \vdots O_{mx(n-m)}]$, we thus have

$$F_1(x_{k+1}) = F_1(x_k) + A s_k = F_1(x_k) - C_k B_k^{-1} F(x_k)$$

$$= F_1(x_k) - F_1(x_k) = 0 . \blacksquare$$

Theorem 3.5 Let A, b, F, F_1, F_2 be defined as in Theorem 3.4. If $C_k = A$ and B_{k+1} is defined by (2.4) for any value of s_k (and any v_k such that $\langle v_k, s_k \rangle \neq 0$), then $C_{k+1} = A$. Furthermore, if either $s_k = -B_k^{-1} F(x_k)$ or $F_1(x_k) = 0$ and $s_k = -\lambda_k B_k^{-1} F(x_k)$, then $F_1(x_{k+1}) = 0$. ∎

Theorem 3.5 shows that once we have correctly obtained the linear part of the Jacobian as Theorem 3.4 shows we are likely to do in n iterations, then our quasi-Newton algorithm will not disturb this information; and whenever we take a quasi-Newton step of length one, which in practice we usually do on our final iterations, we will locate a zero of the linear functions.

4. Local Q-Superlinear Convergence on Nonlinear Problems

In this section we show, subject to reasonable conditions on the function $F : \mathbb{R}^n \to \mathbb{R}^n$, that if x_o is close enough to x^* and if B_o is close enough in norm to $F'(x^*)$ [or $F'(x_o)$], then the sequence of x_i's generated by Algorithm II with $s_i = -B_i^{-1} F(x_i)$ converges Q-superlinearly to x^*. Our proof leans heavily on the local superlinear convergence proof of Broyden, Dennis, and Moré [1973] for Broyden's method; and on the work of Dennis and Moré [1974] characterizing superlinear convergence.

In Theorem 4.2, we give a general condition under which a quasi-Newton algorithm of form (2.1) with steplength one will achieve linear convergence. This theorem amounts to Theorem 3.2 in Broyden, Dennis, Moré [1973] extended to updates using information from previous iterations. Lemmas 4.3 and 4.4 show that the update of Algorithm II satisfies the conditions of Theorem 4.2 along with some further conditions. Using this we show in Theorem 4.5 that Algorithm II achieves local Q-superlinear convergence. We first state a simple lemma which we will use several times; its proof follows immediately from §3.2.5 of [Ortega & Rheinboldt, 1970].

Lemma 4.1 Let $F: \mathbb{R}^n \to \mathbb{R}^n$ be differentiable in the open convex set D, and suppose for some $x^* \in D$ and $\rho > 0$, $K \geq 0$ that

$$||F'(x) - F'(x^*)|| \leq K ||x - x^*||^{\rho} . \tag{4.1}$$

Then for $u, v \in D$,

$$||F(v) - F(u) - F'(x^*)(v-u)|| \leq K||v-u||\max \{||v-x^*||^{\rho}, ||u-x^*||^{\rho}\}. \blacksquare \tag{4.2}$$

Theorem 4.2 Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be differentiable in the open convex set D, and assume for some $x^* \in D$ and $\rho > 0$, $K \geq 0$ that (4.1) holds, where $F(x^*) = 0$ and $F'(x^*)$ is nonsingular. Let $J = F'(x^*)$. Consider sequences $\{x_0, x_1, \ldots\}$ of points in $\mathbb{R}^n$ and $\{B_0, B_1, \ldots\}$ of nonsingular matrices which satisfy

$$x_{k+1} = x_k - B_k^{-1} F(x_k) \tag{4.3}$$

and

$$\begin{aligned} ||B_{k+1} - J||_F \leq ||B_k - J||_F + \alpha \max \{&||x_{k+1} - x^*||^\rho, \\ &||x_k - x^*||^\rho, \ldots, \\ &||x_{k-q} - x^*||^\rho\}, \end{aligned} \tag{4.4}$$

$k = 0,1,\ldots$, for some fixed $\alpha \geq 0$ and $q \geq 0$, where $x_j = x_0$ for $j < 0$. Then for each $r \in (0,1)$, there are positive constants $\varepsilon(r)$, $\delta(r)$ such that if $||x_0 - x^*|| \leq \varepsilon(r)$ and $||B_0 - J||_F \leq \delta(r)$, then the sequence $\{x_0, x_1, \ldots\}$ is well-defined and converges to x^* with

$$||x_{k+1} - x^*|| \leq r||x_k - x^*||$$

for all $k \geq 0$. Furthermore, $\{||B_k||\}$ and $\{||B_k^{-1}||\}$ are uniformly bounded.

The proof is so similar to that of Theorem 3.2 of [Broyden, Dennis, & Moré, 1973] that we omit it. ∎

In Lemma 4.3 we show that for $\hat{s}_i$, $\hat{y}_i$ defined in Algorithm II, asymptotically $||\hat{y}_i - F'(x^*)\,\hat{s}_i||$ is small relative to $||s_i||$. This is the key to proving in Lemma 4.4 that the update of Algorithm II satisfies equation (4.4) of Theorem 4.2.

Lemma 4.3 Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be differentiable in the open convex set D and assume for some $x^* \in D$ and $\rho > 0$, $K \geq 0$ that (4.1) holds, where $F(x^*) = 0$ and $J \equiv F'(x^*)$ is non-singular. Consider the sequences $\{x_0, x_1, \ldots\}$ of points in $\mathbb{R}^n$ and $\{B_0, B_1, \ldots\}$ of nonsingular matrices in $\mathbb{R}^{nxn}$ generated from (x_0, B_0) by Algorithm II with $s_i = -B_i^{-1} F(x_i)$ for all i. Let $\hat{s}_i$ be defined as in Algorithm II and $\hat{y}_i$ as in Theorem 2.3. Then

$$||\hat{y}_i - J\,\hat{s}_i|| \leq \max\{1, \tau^{i-\ell_i-1}\}\, 2^{i-\ell_i} K\, ||s_i||\, m_i, \text{ where} \quad (4.5a)$$

$$m_i = \max\{||x_{\ell_i} - x^*||^\rho, \ldots, ||x_i - x^*||^\rho, ||x_{i+1} - x^*||^\rho\}\,. \quad (4.5b)$$

Proof. The proof is by induction. For $i = 0$, $\hat{s}_0 = s_0$ and $\hat{y}_0 = y_0$, so $||\hat{y}_0 - J\,\hat{s}_0|| = ||y_0 - J\,s_0||$, which is $\leq K||s_0||m_0$ by Lemma 4.1 with $v = x_1$, $u = x_0$. Thus for $i = 0$ (4.5) is true, since $\ell_0 = 0$ by Algorithm II.

Now assume (4.5) holds for $i = 0, \ldots, k-1$. For $i = k$, if $k = \ell_k$, then $\hat{y}_k = y_k$, $\hat{s}_k = s_k$, and $||\hat{y}_k - J\,\hat{s}_k|| \leq K||s_k||m_k$ by Lemma 4.1, so we are done. If $k > \ell_k$, then

$$\begin{aligned}
\hat{y}_k - J\,\hat{s}_k &= y_k - E_k Q_k s_k - J\,s_k + J\,Q_k\,s_k \\
&= (y_k - J\,s_k) - (B_k - J)\,Q_k\,s_k \\
&= (y_k - J\,s_k) - \sum_{j=\ell_k}^{k-1} (B_k - J)\,\hat{s}_j \frac{\langle \hat{s}_j, s_k\rangle}{\langle \hat{s}_j, \hat{s}_j\rangle} \\
&= (y_k - J\,s_k) - \sum_{j=\ell_k}^{k-1} (\hat{y}_j - J\,\hat{s}_j) \frac{\langle \hat{s}_j, s_k\rangle}{\langle \hat{s}_j, \hat{s}_j\rangle}\,,
\end{aligned}$$

the last equation following from $B_k \hat{s}_j = \hat{y}_j$ in Theorem 2.3. Therefore

$$||\hat{y}_k - J \hat{s}_k|| \leq ||y_k - J s_k|| + \sum_{j=\ell_k}^{k-1} ||\hat{y}_j - J \hat{s}_j|| \; ||s_k||/||\hat{s}_j||.$$

Thus, using Lemma 4.1, induction hypothesis 4.5, (2.11f), and the fact that $m_k \geq m_i$, $i = \ell_k, \ldots, k-1$, by the definition of m_i, we have

$$||\hat{y}_k - J \hat{s}_k|| \leq K||s_k||m_k + K||s_k||m_{\ell_k}$$

$$+ \sum_{j=\ell_k+1}^{k-1} \tau^{j-\ell_k-1} 2^{j-\ell_k} K \frac{||s_j||}{||\hat{s}_j||} ||s_k||m_j$$

$$\leq K||s_k||m_k\{1 + 1 + \sum_{j=\ell_k+1}^{k-1} (2\tau)^{j-\ell_k}\}$$

$$\leq K||s_k||m_k \tau^{k-\ell_k-1} \{1 + \sum_{j=\ell_k}^{k-1} 2^{j-\ell_k}\}$$

$$= K||s_k||m_k \tau^{k-\ell_k-1} 2^{k-\ell_k},$$

which proves (4.5) for $i = k$ and completes the induction. ∎

Lemma 4.4 Let all the conditions of Lemma 4.3 hold. Then

$$||B_{i+1} - J||_F \leq ||B_i - J||_F \sqrt{1 - \Theta_i^2} + (2\tau)^{n-1} K m_i, \tag{4.6a}$$

where

$$\Theta_i = \frac{||(B_i - J) \hat{s}_i||}{||B_i - J||_F \, ||\hat{s}_i||}. \tag{4.6b}$$

Proof: Using the definitions of $\hat{s}_i$ and $\hat{y}_i$ along with the equation $\langle \hat{s}_i, s_i \rangle = \langle \hat{s}_i, \hat{s}_i \rangle$ from Theorem 2.3, we find

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i)\, \hat{s}_i^T}{\hat{s}_i^T s_i} = B_i + \frac{(\hat{y}_i - B_i \hat{s}_i)\, \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} .$$

Therefore

$$B_{i+1} - J = (B_i - J) \left[I - \frac{\hat{s}_i \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} \right] + \frac{(\hat{y}_i - J \hat{s}_i)\, \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} , \text{ and}$$

$$\|B_{i+1} - J\|_F \leq \left\| (B_i - J) \left[I - \frac{\hat{s}_i \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} \right] \right\|_F + \left\| \frac{(\hat{y}_i - J \hat{s}_i)\, \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} \right\|_F . \tag{4.7}$$

Broyden, Dennis, and Moré [1973] show that for $E \in \mathbb{R}^{n \times n}$ and $u \in \mathbb{R}^n$,

$$\|E \left[I - \frac{u u^T}{u^T u} \right]\|_F^2 = \|E\|_F^2 - \frac{\|Eu\|^2}{\|u\|^2} . \quad \text{Thus}$$

$$\|(B_i - J) \left[I - \frac{\hat{s}_i \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} \right]\|_F^2 = \|B_i - J\|_F^2 \left[1 - \frac{\|(B_i - J) \hat{s}_i\|^2}{\|B_i - J\|_F^2 \, \|\hat{s}_i\|^2} \right] . \tag{4.8a}$$

Secondly,

$$\left\| \frac{(\hat{y}_i - J \hat{s}_i)\, \hat{s}_i^T}{\hat{s}_i^T \hat{s}_i} \right\|_F = \frac{\|\hat{y}_i - J \hat{s}_i\|}{\|\hat{s}_i\|} \tag{4.8b}$$

$$\leq \max \{1, \tau^{i-\ell_i-1}\} K \frac{\|s_i\|}{\|\hat{s}_i\|} 2^{i-\ell_i} m_i \leq (2\tau)^{n-1} K m_i$$

from (2.11 f-g) and Lemma 4.3. Combining (4.7-8) gives (4.6). ∎

Lemma 4.4 shows that Algorithm II satisfies the conditions of Theorem 4.2 and is locally linearly convergent for any $r \in (0,1)$. The extra power supplied by the $\sqrt{1 - \theta_i^2}$ term in equation (4.6) enables us to prove local Q-superlinear convergence.

Theorem 4.5 Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be differentiable in the open convex set D, and assume for some $x^* \in D$ and $\rho > 0$, $K \geq 0$, that (4.1) holds, where $F(x^*) = 0$ and $J \equiv F'(x^*)$ is nonsingular. Consider the sequence $\{x_0, B_0, x_1, B_1, x_2, B_2, \ldots\}$, $x_i \in \mathbb{R}^n$, $B_i \in \mathbb{R}^{n \times n}$, generated from (x_0, B_0) by Algorithm II with $s_i = -B_i^{-1} F(x_i)$ for all i. Then there exist $\varepsilon, \delta > 0$ such that for $||x_0 - x^*|| \leq \varepsilon$ and $||B_0 - J||_F \leq \delta$, $\{x_i\}$ converges Q-superlinearly to x^* and $\{||B_k||\}$, $\{||B_k^{-1}||\}$ are bounded.

Proof: The linear convergence of Algorithm II, and boundedness of $\{||B_i||\}$, $\{||B_i^{-1}||\}$, follow Theorem 4.2 and Lemma 4.4. The term $(2\tau)^{n-1}K$ in (4.6) corresponds to α in (4.4).

We turn now to the superlinear convergence of Algorithm II. From Lemma 4.4 we have

$$||B_{i+1} - J||_F \leq ||B_i - J||_F \sqrt{1 - \theta_i^2} + \alpha m_i, \quad \text{where} \tag{4.9a}$$

$$\theta_i = \frac{||(B_i - J)\,\hat{s}_i||}{||B_i - J||_F\, ||\hat{s}_i||} . \tag{4.9b}$$

If $\liminf \{||B_i - J||_F\} = 0$, then Corollary 3.3 of Broyden, Dennis and Moré [1973] shows that Algorithm II is Q-superlinearly convergent.

Now suppose lim inf $\{||B_i - J||_F\} > 0$. From the linear convergence of Algorithm II we know $\lim m_i = 0$. By (4.9) we must therefore have $\lim \theta_i^2 = 0$, i.e.,

$$\lim_{i\to\infty} \frac{||(B_i - J)\ \hat{s}_i||}{||\hat{s}_i||} = 0 \quad . \tag{4.10}$$

Now Theorem 2.2 of Dennis and Moré [1974] shows, under the conditions of Theorem 4.5, that if Algorithm II is linearly convergent, then

$$\lim_{i\to\infty} \frac{||(B_i - J)\ s_i||}{||s_i||} = 0 \tag{4.11}$$

is a sufficient (and necessary) condition for local Q-superlinear convergence of the algorithm. Therefore it only remains to show that (4.10) implies (4.11).

Let

$$\hat{Q}_i = \sum_{j=\ell_i}^{i-1} \frac{\hat{s}_j\ \hat{s}_j^T}{\hat{s}_j^T\ \hat{s}_j} \quad , \qquad \text{so that}$$

$\hat{s}_i = (I - \hat{Q}_i)\ s_i$. Now $||I - \hat{Q}_i|| = 1$ because $(I - \hat{Q}_i)$ is a non-zero orthogonal projection matrix, so $||\hat{s}_i|| \leq ||s_i||$ and

$$\frac{||(B_i - J)\ \hat{s}_i||}{||\hat{s}_i||} \geq \frac{||(B_i - J)\ \hat{s}_i||}{||s_i||} \quad . \tag{4.12}$$

By the triangle inequality,

$$\frac{||(B_i - J)\ s_i||}{||s_i||} \leq \frac{||(B_i - J)\ \hat{s}_i||}{||s_i||} + \frac{||(B_i - J)\ \hat{Q}_i\ s_i||}{||s_i||} \quad . \tag{4.13}$$

As $i\to\infty$, the first term on the right hand side of (4.13) approaches zero due to (4.10), (4.12). For the second term on the right side of (4.13), Theorem 2.3 and Lemma 4.3 show

$$\begin{aligned}||(B_i - J)\,\hat{Q}_i\, s_i|| &= \Big|\Big|\sum_{j=\ell_i}^{i-1} (B_i - J)\,\hat{s}_j\ \langle\hat{s}_j, s_i\rangle/\langle\hat{s}_j, \hat{s}_j\rangle\Big|\Big| \\ &= \Big|\Big|\sum_{j=\ell_i}^{i-1} (\hat{y}_j - J\,\hat{s}_j)\,\langle\hat{s}_j, s_i\rangle/\langle\hat{s}_j, \hat{s}_j\rangle\Big|\Big| \\ &\le \sum_{j=\ell_i}^{i-1} \max\{1, \tau^{j-\ell_i-1}\}\, 2^{j-\ell_i}\, K\, \frac{||s_j||}{||\hat{s}_j||} \\ &\qquad \cdot ||s_i||m_j\ .\end{aligned}$$

Because $||s_j||/||\hat{s}_j|| \le \tau$ (by (2.11f)) and $m_j \le m_{i-1}$, $j=\ell_i,\ldots,i-1$ with $i-\ell_i < n$ (by (2.11g)), we thus have

$$\begin{aligned}||(B_i - J)\,\hat{Q}_i\, s_i|| &\le K||s_i||m_{i-1}\,\tau^{i-\ell_i-1}\sum_{j=\ell_i}^{i-1} 2^{j-\ell_i} \\ &\le K||s_i||\,\tau^{i-\ell_i-1}\,2^{i-\ell_i}\,m_{i-1} \\ &\le K||s_i||\,\tau^{n-2}\,2^{n-1}\,m_{i-1}\ .\end{aligned}$$

Hence

$$\frac{||(B_i - J)\,\hat{Q}_i\, s_i||}{||s_i||} \le K\,\tau^{n-2}\,2^{n-1}\,m_{i-1}\ , \quad \text{so}$$

$$\lim_{i\to\infty} \frac{||(B_i - J)\,\hat{Q}_i\, s_i||}{||s_i||} = 0\ , \tag{4.14}$$

since $\lim_{i\to\infty} m_i = 0$. Therefore (4.10) and (4.12-14) imply (4.11) is true, which completes the proof of local Q-superlinear convergence of Algorithm II. ∎

It should be noted that the techniques of this section apply equally well to an algorithm identical to II except restarting whenever $i - \ell_{i-1} \geq t$, $t < n$ (or $||s_i||/||\hat{s}_i|| > \tau$). Such an algorithm would not be exact on linear problems, however. Another interesting algorithm covered by the techniques of this section is one setting

$$\hat{s}_i = s_i - s_{i-1} \frac{\langle s_{i-1}, s_i \rangle}{\langle s_{i-1}, s_{i-1} \rangle}$$

at each iteration. Such an algorithm would preserve the current and most recent quasi-Newton equation at each step, and can be shown by the techniques of this section to be Q-superlinearly convergent without restarts. We have not tested this algorithm.

Finally, the techniques of this section would also apply to an algorithm which set $\hat{s}_i$ equal to the projection of s_i orthogonal to the previous t s_j's, $t < n$, subject to the strong linear independence of $s_{i-t}, \ldots, s_i$ as in Algorithm II. Such an algorithm would require no restarts and would be exact for linear problems if $t = n$. It would be fairly easy to implement (in $O(n^2)$ housekeeping operations per step) using Powell's [1968] orthogonalization scheme.

5. Computational Results

We have implemented Algorithms II and II', with some modifications, and tested them on several problems. In Step (2.10a) we choose $s_i = -\lambda_i B_i^{-1} F(x_i)$, where λ_i is determined by the scheme described in [Broyden, 1965] with the added restriction that $||s_i||_\infty \leq 1$ (except as otherwise noted). Instead of storing B_i, we actually store and update $H_i = B_i^{-1}$. Rather than compute Q_i explicitly by formula (2.10c), we use appropriate Householder transformations to express in product form an orthogonal matrix P_i such that

$$Q_i = P_i^T \begin{bmatrix} I_{i-\ell_{i-1}} & 0 \\ 0 & 0 \end{bmatrix} P_i \text{ , whence } s_i - Q_i s_i = P_i^T \begin{bmatrix} 0 & 0 \\ 0 & I_{n-i+\ell_{i-1}} \end{bmatrix} P_i s \text{ .}$$

Our implementation includes the option suggested above of restarting whenever $i-\ell_{i-1} \geq t$, where $t \leq n$ is fixed. For $t = 1$ this lets us try Broyden's original methods on the test problems.

Test Problems

The test problems we used include the following; we write x^i for the i^{th} component of $x = (x^1, \ldots, x^n)^T \in \mathbb{R}^n$.

Problem 1 [Brown, 1969, p. 567]: $n = 5$.

$$f_i(x) = -(n+1) + 2x^i + \sum_{\substack{j=1 \\ j \neq i}}^{n} x^j, \quad 1 \leq i \leq n-1 \text{ .}$$

$$f_n(x) = -1 + \prod_{j=1}^{n} x^j \text{ .}$$

$$x_0 = (.5, .5, \ldots, .5)^T; \quad x^* = (1, 1, \ldots, 1)^T \text{ .}$$

Problem 2 [Brown, 1969, p. 567]: $n = 2$.

$$f_1(x) = (x^1)^2 - x^2 - 1 .$$

$$f_2(x) = (x^1-2)^2 + (x^2 - .5)^2 - 1 .$$

$$x_0 = (.1,2); \; x^* \doteq (1.06735, .139228)^T .$$

Problem 3 - "Chebyquad" - [Fletcher, 1965, p. 36]: $n = 2,3,4,5,6,7,9$.

$f_i(x) = \int_0^1 T_i(\zeta)d\zeta - \frac{1}{n}\sum_{j=1}^{n} T_i(x^j)$, where T_i is the i^{th} Chebyshev polynomial, transformed to the interval $[0,1]$, i.e. $T_0(\zeta) = 1$, $T_1(\zeta) = 2\zeta - 1$, $T_{i+1}(\zeta) = 2(2\zeta-1)T_i(\zeta) - T_{i-1}(\zeta)$ for $i \geq 1$. Note that

$$\int_0^1 T_i(\zeta)d\zeta = \begin{cases} 0 \text{ if } i \text{ is odd} \\ -1/(i^2-1) \text{ if is is even.} \end{cases}$$

$x_0^j = j/(n+1)$, $1 \leq j \leq n$; the components x^{*j} of a solution are any permutation of the abscissae for the Chebyshev quadrature rule of order n.

None of the variations of Broyden's method which we tried solved this problem for $n=9$, so we omit the results of these runs.

Problem 4 [Brown and Conte, 1967]: $n = 2$.

$$f_1(x) = \frac{1}{2}\sin(x^1x^2) - \frac{x^2}{4\pi} - \frac{x^1}{2} .$$

$$f_2(x) = (1 - \frac{1}{4\pi})[\exp(2x^1) - e] + \frac{ex^2}{\pi} - 2ex^1 .$$

$$x_0 = (.6,3)^T; \; x^* = (.5,\pi)^T .$$

Problem 5 [Brown and Gearhart, 1971, p. 341]: $n = 3$.

$$f_1(x) = (x^1)^2 + 2(x^2)^2 - 4 .$$

$$f_2(x) = (x^1)^2 + (x^2)^2 + x^3 - 8 .$$

$$f_3(x) = (x^1-1)^2 + (2x^2 - \sqrt{2})^2 + (x^3-5)^2 - 4.$$

$$x_0 = (1, .7, 5)^T; \; x^* = (0, \sqrt{2}, 6)^T .$$

Problem 6 [Deist and Sefor, 1967]: $n = 6$.

$$f_i(x) = \sum_{\substack{j=1 \\ j\neq 1}}^{6} \cot \beta_i x^j, \qquad 1 \leq i \leq 6, \quad \text{where}$$

$$(\beta_1,\ldots,\beta_6) = 10^{-2} (2.249, 2.166, 2.083, 2, 1.918, 1.835) .$$

$$x_o = (75, 75,\ldots,75)^T; \; x^* \doteq (121.850, 114.161, 93.6488, 62.3186, 41.3219, 30.5027)^T .$$

Problem 7 [Broyden, 1965]: $n = 5, 10$.

$$f_1(x) = (.5x^1-3)x^1 + 2x^2 - 1.$$

$$f_i(x) = x^{i-1} + (.5x^i-3)x^i + 2x^{i+1} - 1, \; 2 \leq i \leq n-1 .$$

$$f_n(x) = x^{n-1} + (.5x^n-3)x^n - 1.$$

$$x_0 = (-1, -1,\ldots,-1)^T .$$

For $n = 5$, $x^* \doteq (-.968354, -1.18696, -1.14848, -.958989, -.594159)^T$ and for $n = 10$, $x^* \doteq (-1.03011, -1.31044, -1.37992, -1.39071, -1.37963, -1.34993, -1.29066, -1.17748, -.967501, -.596526)^T$.

We ran our tests in double precision on the IBM 370/168 at Cornell University. Table I below gives the results of some of these tests. "Problem $\alpha.\nu$" means problem α with $n = \nu$.

For each test problem we report both the actual number of function evaluations needed to achieve $||F|| < 10^{-10}$ and a normalized number of function evaluations obtained by dividing the actual number by the minimum of the three numbers for that problem (and rounding to two decimal places). Although Algorithm II sometimes fares worse then Broyden's good method, the means of the normalized numbers show that Algorithm II with $\tau = 10$ averaged about 10% fewer function evaluations than Broyden's good method on these test problems. The choice $\tau = 10$ worked considerably better than $\tau = 100$ in Algorithm II, suggesting that a reasonably small value of τ, such as 10, may be best.

We ran several other tests, which we shall not report in detail. True to its name, for example, Broyden's bad method failed six times as often as his good method. Algorithm II' with $\tau = 10$ failed on 5 of the 15 test runs; with $\tau = 100$ it failed on only 3, but fared rather worse than Broyden's good method with respect to mean normalized function evaluations. We tried a hybrid between Algorithms II and II' whose average behavior for $\tau = 10$ was as good as that of Algorithm II. The hybrid applies the projections of Algorithm II' to the inverse form of Broyden's good method, so that $y_i - Q_i'y_i$ is replaced by $(I - Q_i')H_i^T s_i$ and the choice $\hat{y}_i = y_i$ is replaced by $\hat{y}_i = H_i^T s_i$.

Problem	Total Function Evaluations			Normalized Function Evaluations		
	Broyden's "Good"	Algorithm II $\tau = 10$	$\tau = 100$	Broyden's "Good"	Algorithm II $\tau = 10$	$\tau = 100$
1.5	31	27	28	1.15	1.00	1.04
2.2	11	10	10	1.10	1.00	1.00
3.2	9	9	9	1.00	1.00	1.00
3.3	13	11	13	1.18	1.00	1.18
3.4	19	23	23	1.00	1.21	1.21
3.5	20	24	23	1.00	1.20	1.15
3.6	(31)[1]	26	33	--	1.00	1.27
3.7	45	35	36	1.29	1.00	1.03
4.2	12	10	10	1.20	1.00	1.00
5.3	15	(28)[1]	(28)[1]	1.00	--	--
5.3[2]	16	15	15	1.07	1.00	1.00
6.6[3]	62	29	60	2.14	1.00	2.07
6.6[2]	32	28	57	1.14	1.00	2.04
7.5	13	13	13	1.00	1.00	1.00
7.10	21	20	20	1.05	1.00	1.00
			Mean	1.17	1.03	1.21
			Std.Dev	.29	.074	.37
			Failures	1	1	1

Table I: Function Evaluations Required to Achieve $||F|| < 10^{-10}$

Notes:

1. Broyden's [1965] quadratic interpolation technique failed to reduce $||F||$ in 10 function evaluations. The number reported is the total number of function evaluations at the time of failure.
2. $||F||$ was allowed to increase as much as twofold (per step) and a maximum steplength of 10 rather than 1 was allowed.
3. A maximum steplength $||s_i||_\infty$ of 10 rather than 1 was allowed.

6. Summary and Conclusions

We have introduced some new quasi-Newton algorithms for solving systems of n non-linear equations in n unknowns. These methods are modifications of "Broyden's good method" and "Broyden's bad method" (Broyden [1965]). They retain the local Q-superlinear convergence of the unmodified methods and have the additional property that if any of the equations are linear, then the methods locate a zero of these equations in n+1 or fewer iterations. (We have only proven these properties in this paper for the modified Broyden's good method, but virtually the same proofs go through for the modified bad method.)

Our computational results suggest that our modified form of Broyden's good method performs better, on the average, than the original form. We think our new method should be further tested and possibly considered as a replacement for the conventional Broyden's method in existing subroutines.

Acknowledgement

We are grateful to Professor M.J.D. Powell for helpful discussions and advice.

7. References

Brown, K.M. (1969), "A Quadratically Convergent Newton-Like Method Based Upon Gaussian Elimination," SIAM J. Numer. Anal. 6, pp. 560-569.

Brown, K.M., & Conte, S.D. (1967), "The Solution of Simultaneous Nonlinear Equations," Proc. 22nd National Conference of the ACM, Thompson Book Co., Washington, D.C., pp. 111-114.

Brown, K.M., & Gearhart, W.B. (1971), "Deflation Techniques for the Calculation of Further Solutions of a Nonlinear System," Numer. Math. 16, pp. 334-342.

Broyden, C.G. (1965), "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comput. 19, pp. 577-593.

Broyden, C.G.; Dennis, J.E.; & Moré, J.J. (1973), "On the Local and Superlinear Convergence of Quasi-Newton Methods," J. Inst. Math. Appl. 12, pp. 223-245.

Davidon, W.C. (1975), "Optimally Conditioned Optimization Algorithms Without Line Searches," Math. Programming 9, pp. 1-30.

Deist, F.H.; & Sefor, L. (1967), "Solution of Systems of Nonlinear Equations by Parameter Variation," Comput. J. 10, pp. 78-82.

Dennis, J.E., Jr.; & Moré, J.J. (1974), "A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods," Math. Comput. 28, pp. 549-560.

Dennis, J.E., Jr.; & Moré, J.J. (1977), "Quasi-Newton Methods, Motivation and Theory," SIAM Rev. 19, pp. 46-89.

Fletcher, R. (1965), "Function Minimization Without Evaluating Derivatives; a Review," Comput. J. 8, pp. 33-41.

García-Palomares, U.M. (1973), "Superlinearly Convergent Quasi-Newton Methods for Nonlinear Programming," Ph.D. dissertation, University of Wisconsin.

Gay, D.M. (1977), "Convergence Properties of Broyden-Type Methods on Linear Systems of Equations," in preparation.

Ortega, J.M.; & Rheinboldt, W.C. (1970), Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York.

Powell, M.J.D. (1968), "On the Calculation of Orthogonal Vectors," Comput. J. 11, pp. 302-304.

Powell, M.J.D. (1976), "Quadratic Termination Properties of Davidon's New Variable Metric Algorithm," Math. Programming, (to appear).

Schnabel, R.B. (1977), Ph.D. Thesis, Computer Science Dept., Cornell University.