BROWN'S METHOD AND SOME GENERALIZATIONS, WITH APPLICATIONS TO MINIMIZATION PROBLEMS+

David M. Gay

TR 75-225

January 1975

Department of Computer Science
Cornell University
Ithaca, New York 14853

+This research was supported in part by National Science Foundation Grant GJ-27528 and GJ40903.

Biographical Sketch

David M. Gay was born on 27 February 1949 in Detroit, Michigan. At the age of four he moved with his parents to Alma, Michigan, where he subsequently attended public school, graduating as co-valedictorian from Alma High School in June, 1967. In August of that year he entered the University of Michigan in Ann Arbor, his senior year at which was spent in Freiburg im Breisgau, West Germany, at the Albert-Ludwigs-Universität; he was inducted into Phi Beta Kappa at the end of his junior year and graduated "with High Distinction, Highest Honors in Mathematics" in August, 1971. That same month he entered the Graduate School of Cornell University, and in August of 1973 he was conferred a Master of Science degree in Computer Science. His work experience includes three summers as a programmer for Leonard Refineries, Inc. (now TOTAL Leonard, Inc.) in Alma, Michigan. During his third year at Cornell, he wrote two technical reports dealing with Scolnik's approach to linear programming. He is a member of the Association for Computing Machinery, as well as SIGMAP and SIGNUM.

Dedication

This thesis is dedicated to my family, especially my parents, without whose encouragement and moral support this work would never have been possible, and to my home town, Alma, Michigan, "Scotland, U.S.A."

Acknowledgments

I would like to express sincere and hearty thanks to Professor John E. Dennis, Jr., for the guidance he gave me as my advisor and chairman of my Special Committee. I would also like to thank Professors Mark J. Eisner and John H. Williams for serving on my Special Committee. In addition, I am grateful to the National Science Foundation, which supported me for two years with a Graduate Fellowship. Finally, I wish to thank Mrs. Dorothy Baker for her careful typing of the manuscript.

This work was partially supported by NSF Grant GJ40903.

BROWN'S METHOD AND SOME GENERALIZATIONS, WITH APPLICATIONS TO MINIMIZATION PROBLEMS

David M. Gay, Ph.D.
Cornell University 1975

Newton's method attempts to find a zero of $f \in C^1(\mathbb{R}^n)$ by taking a step which is intended to make all components of f vanish at once. In this respect Newton's method processes the components of f in parallel. Contrasting to this, Brown's method and the generalizations thereof considered in this thesis process the components of f serially, one after another. One major iteration of these methods may be described as follows: given the starting point (i.e. current major iterate) y_0, linearize the first component f_1 of f at y_0 and find a point y_1 in the $n-1$ dimensional hyperplane H_1 on which this linearization vanishes; in general, having found a point y_k $(1 \le k < n)$ in the $n-k$ dimensional hyperplane H_k on which the heretofore constructed linearizations vanish, restrict f_{k+1} to H_k, linearize this restriction at y_k, and find a point y_{k+1} in the $n-(k+1)$ dimensional hyperplane H_{k+1} on which this linearization vanishes; stop when y_n has been found and let y_n be the next major iterate. When f is a general nonlinear function and finite differences are used to construct the linearizations, this approach must do work equivalent to approximating only about half the components of f' and thus requires only about half as many function evaluations per major iteration as the corresponding finite difference Newton's method, while still enjoying the same rate of local convergence.

The view we take of Brown's method differs from the one he had in deriving his method. Our view, suggested in a paper by Brent, leads us to a more efficient implementation of Brown's method and lets us see the connection between it, Brent's variation thereof, Craig's method for solving linear equations, and Stewart's generalized conjugate direction algorithms. Furthermore, this view leads us to propose new descent algorithms for unconstrained and constrained minimization problems. One of these algorithms is closely related to some generalized conjugate gradient algorithms and requires about half as many function evaluations per major iteration (reinitialization plus n steps) as what may be the most reasonable corresponding finite difference version of these conjugate gradient methods, while retaining the same rate of convergence, albeit with increased housekeeping overhead.

Of the minimization methods we propose, the ones requiring the fewest function evaluations per major iteration need about the same number as the finite difference Newton's method (with $f = \nabla\Phi$, where Φ is to be minimized) and have the same rate of convergence. They therefore appear locally to have no particular advantage over the latter method. However, since our approach processes the rotated components of $\nabla\Phi$ serially, we may terminate a major iteration early if the effort needed to complete it does not appear warranted. Under some conditions our approach might thus require less effort in terms of function evaluations than the finite difference Newton's method.

Preliminary numerical tests suggest that our approach to unconstrained minimization holds promise and that more experimentation is called for. We suggest several modifications which might improve performance.

Since inexact arithmetic is normally used in practice, we discuss using finite but variable precision arithmetic in some of the algorithms without disturbing their rate of convergence. We also mention some considerations for the usual case when only finite, fixed precision arithmetic is available.

CONTENTS

1. Preliminaries

1.1 Introduction

Frequently situations arise in which a system of n nonlinear equations in n unknowns must be approximately solved. Thus, given a mapping

$$f: \mathbb{R}^n \to \mathbb{R}^n$$

from Euclidean n-space into itself, we seek a point $x \in \mathbb{R}^n$ at which $f(x) \approx 0$.

Assume that f is twice continuously differentiable and that $x^* \in \mathbb{R}^n$ is a point where $f(x^*) = 0$ and the Jacobian matrix $f'(x^*)$ of f at x^* is nonsingular. Many iterative techniques have been devised which, given an initial guess $x_0 \in \mathbb{R}^n$ sufficiently close to x^*, produce a sequence $x_1, x_2, \ldots$ which converges to x^*. Most of these techniques work with linearizations of f. Given x_i, many of them compute a linearization of the form

$$f(x) \approx f(x_i) + A^{(i)}(x-x_i),$$

where $A^{(i)}$ is an $n \times n$ matrix, then obtain x_{i+1} by solving the linear system $A^{(i)}(x_{i+1}-x_i) = -f(x_i)$. For example, Newton's method chooses $A^{(i)} = f'(x_i)$, while the various quasi-Newton methods, a number of which are described in Section 2 of (Broyden, et al., 1973), choose $A^{(i)}$ to be $A^{(i-1)}$ plus a rank 1 or rank 2 correction matrix such that $f(x_i) = f(x_{i-1}) + A^{(i)}(x_i-x_{i-1})$.

Under suitable hypotheses, Newton's method enjoys local Q-quadratic convergence (Ortega and Rheinboldt, 1970, p. 312): if $\|\cdot\|$ denotes a norm on

$\mathbb{R}^n$ and $\|x_0-x^*\|$ is sufficiently small, then $\|x^{i+1}-x^*\| \leq c\|x^i-x^*\|^2$ for all $i \geq 0$ and some $c \in \mathbb{R}$ which does not depend on k. However, Newton's method uses exact analytic derivatives, which may be difficult or impossible to obtain in practice. This disadvantage may be overcome by replacing $f'(x_i)$ by a finite difference approximation; Shamanskii (1967) (see also (Dennis, 1970)) has shown that this approximation may be chosen so that the resulting finite difference Newton's method is still locally Q-quadratically convergent.

In practice, function evaluations usually require considerably more time than do the remaining calculations and overhead in a nonlinear equations solving algorithm. The finite difference Newton's method requires (at least) n^2+n component function evaluations per iteration, whereas the quasi-Newton methods need but n+1. However, the latter may not make as much progress towards the solution in one iteration as the former. To be sure, Broyden, Dennis, and Moré (1973) have shown, under reasonable conditions, that single- and double-rank quasi-Newton methods are locally Q-superlinearly convergent; moreover, for the unconstrained minimization problem, in which a local minimizer of a twice continuously differentiable $\phi: \mathbb{R}^n \to \mathbb{R}$ is sought with $f = \nabla\phi$, the work of Burmeister (1973), and Dixon (1972) shows, under appropriate conditions, that various double-rank quasi-Newton methods display local n-step quadratic convergence. But for the latter problem, Burmeister (1973) indicates that the Ostrowski efficiency index (Ostrowski, 1960, §3.11), measured with respect to function evaluations, may be greater for a certain multistep finite difference Newton's method than for the Davidon-Fletcher-Powell algorithm (one of the good quasi-Newton methods). And for general f it is not clear which approach is more efficient. Thus for some problems the finite difference Newton's method might require less total work than a quasi-Newton method to obtain a highly accurate approximate solution.

Brown (1966) has suggested a method which, like Newton's method, enjoys local quadratic convergence (Brown and Dennis, 1971), but which requires only $n(n+3)/2$ function evaluations per iteration, or roughly half as many as the finite difference Newton's method. We may view Brown's method as follows. (This view, which allows us in §2.2 to implement the method more efficiently than did Brown (1973), differs from Brown's and was suggested in a paper by Brent (1973). For a slightly different and less detailed statement of this view, see the conclusion, §5.) One after another, Brown's method constructs linear approximations $\bar{F}_j$ to the component functions f_j of $f=(f_1,\ldots,f_n)^T$ and finds a sequence of minor iterates $y_0,y_1,\ldots,y_n$ such that $\bar{F}_j(y_k) = 0$ for $1 \le j \le k$; if this process starts at the major iterate x_{i-1}, that is, if $y_0 = x_{i-1}$, then the final minor iterate y_n becomes the next major iterate x_i, i.e., $x_i = y_n$. To construct $\bar{F}_k$, which is intended to approximate f_k well near y_{k-1}, Brown's method relies on the fact that f_k is continuously differentiable, so that $f_k(y_{k-1})+f'(y_{k-1})(y-y_{k-1})$ furnishes a good linear approximation to $f_k(y)$, at least for y sufficiently close to y_{k-1}. By finite differences, the method implicitly constructs an approximation $\bar{w}_k^T$ to $f_k'(y_{k-1})$ and then defines

$$\bar{F}_k(y) \equiv f_k(y_{k-1}) + \bar{w}_k^T(y-y_{k-1}).$$

Now $\bar{F}_k$ vanishes on a certain hyperplane H_k normal to $\bar{w}_k$. The next minor iterate y_k is chosen to lie in H_k; in fact, in order that $\bar{F}_j(y_k) = 0$ for $1 \le j \le k$, y_k is chosen to be a point in $\bigcap_{j=1}^{k} H_j$. Since $\bar{F}_1(y_{k-1})=\ldots=\bar{F}_{k-1}(y_{k-1}) = 0$, whence $y_{k-1} \in \bigcap_{j=1}^{k-1} H_j$, it is necessary to choose y_k in such a way that $\bar{w}_j^T(y_k-y_{k-1}) = 0$, $1 \le j \le k-1$. Hence Brown's method selects a certain $p_k \in \mathbb{R}^n$ with $\bar{w}_j^T p_k = 0$ for $j < k$ and $\bar{w}_k^T p_k \neq 0$, then chooses λ_k such that $\bar{F}_k(y_{k-1} - \lambda_k p_k) = 0$, i.e., $f_k(y_{k-1}) = \lambda_k \bar{w}_k^T p_k$, and sets $y_k = y_{k-1} - \lambda_k p_k$.

In order to determine y_k from y_{k-1}, Brown's method needs only to find a vector p_k orthogonal to $\bar{w}_1, \ldots, \bar{w}_{k-1}$ such that $\bar{w}_k^T p_k \neq 0$ and then to compute the value of $\bar{w}_k^T p_k$ (since then $y_k = y_{k-1} - \lambda_k p_k$ with $\lambda_k = f_k(y_{k-1})/\bar{w}_k^T p_k$). To do this, the method constructs a certain basis $u_k^{(k)}, \ldots, u_n^{(k)}$ for the orthogonal complement Span $\{\bar{w}_1, \ldots, \bar{w}_{k-1}\}^{\perp}$ of $\bar{w}_1, \ldots, \bar{w}_{k-1}$. By finite differences, approximations $\omega_j^{(k)}$ to $f_k'(y_{k-1}) u_j^{(k)}$ are found, and $\bar{w}_k$ is implicitly chosen so that $\omega_j^{(k)} = \bar{w}_k^T u_j^{(k)}$, $k \leq j \leq n$, and $\bar{w}_k^T \bar{w}_j = f_k'(y_{k-1}) \bar{w}_j$ for $1 \leq j \leq k-1$; then j^* is chosen so that $|\omega_{j^*}^{(k)}| = \max\{|\omega_j^{(k)}| : k \leq j \leq n\}$, thus assuring that $\bar{w}_k^T u_{j^*}^{(k)} \neq 0$, and $u_{j^*}^{(k)}$ is then taken to be p_k. The quantity $\lambda_k = \bar{w}_k^T p_k = \omega_{j^*}^{(k)}$ is thus already at hand.

In this thesis we examine some generalizations of Brown's method which construct the linearizations $\bar{F}_k$ differently or make other choices of the search direction p_k. In particular, we consider the unconstrained minimization problem, in which $f = \nabla\phi$ and we wish to minimize $\phi: \mathbb{R}^n \to \mathbb{R}$. We show how to construct algorithms for solving this problem which, under reasonable conditions, are globally and Q-quadratically convergent. Finally, we indicate how the approach presented here extends to the constrained minimization problem, which requires us to minimize ϕ subject to continuously differentiable equality and inequality constraints; we sketch the construction of algorithms for this problem which, again under reasonable conditions, are globally and Q-quadratically convergent.

The minimization methods which we propose assume that analytic derivatives exist but are not available and compute finite difference approximations to the necessary derivatives. Indeed, if analytic gradients are available, they may be used; but if one analytic gradient evaluation costs as much as $3+(n/4)$ evaluations of the objective function, then it will be cheaper to ignore the analytic gradient and use finite differences, at least as long as sufficiently accurate arithmetic is cheaply available.

Our minimization methods require at least $n^2/2+O(n)$ function evaluations per major iteration, which may sound like a high price to pay for quadratic convergence. However, as Brent (1973) has pointed out for his variant of Brown's method, it is possible to reuse the second derivative information computed during one major iteration in several of the subsequent major iterations; while this lowers the convergence rate per major iteration, it may make the method more efficient.

The algorithms which we consider here may be viewed as generalized conjugate directions algorithms. Consider momentarily the linear system $Ax^* = b$, where A is a symmetric, positive definite $n \times n$ matrix and $b \in \mathbb{R}^n$. A conjugate directions algorithm solves this system by generating the columns $p_1,\ldots,p_n$ of a nonsingular matrix P of search directions such that P^TAP is diagonal, selecting an initical guess y_0 at x^*, and computing

$$r_k = Ay_{k-1} - b,\ y_k = y_{k-1} - \left(\frac{p_k^Tr_k}{p_k^TAp_k}\right)p_k \quad \text{for} \quad 1 \leqq k \leqq n.$$

The residuals r_k satisfy $p_j^TAr_k = 0$ for $1 \leqq j \leqq k$. Thus $r_{n+1}=0$ and $x^*=y_n$. Stewart (1972) pointed out that the vectors p_k serve both as search directions and as conjugating directions. He suggested that for any nonsingular $n \times n$ matrix A, a pair (P,Q) of $n \times n$ matrices be called A-conjugate if Q^TAP is a nonsingular lower triangular matrix. If $p_1,\ldots,p_n$ are the columns of P and $q_1,\ldots,q_n$ are those of Q, then the linear system $Ax^*=b$ may be solved as follows: let y_0 be any initial guess at x^* and compute

$$r_k = Ay_{k-1} - b,\ y_k = y_{k-1} - \left(\frac{q^T_k r_k}{q_k^T A p_k}\right) p_k \quad \text{for } 1 \le k \le n.$$

Note that $q_k^T A p_k$ is nonzero, since it is the k-th diagonal element of the nonsingular, lower triangular matrix $Q^T AP$. We have $q_j^T Ar_k = 0$ for $1 \le j \le k$, so again $r_{n+1} = 0$ and $x^* = y_n$. Here p_k acts as the k-th search direction while q_k acts as the k-th conjugating direction. When applied to the linear $f(x) \equiv Ax - b$, each zero-finding method discussed in §2 constructs an A-conjugate pair (P,Q) and, given y_0, produces precisely the sequence $y_1,\ldots,y_n$ of minor iterates described above. Moreover, when applied to the quadratic $\phi(x) \equiv \frac{1}{2} x^T Ax - b^T x$, where A is symmetric and positive definite, the unconstrained minimization methods proposed in §3 find the zero x^* of the gradient $\nabla\phi(x) = Ax-b$ by choosing A-conjugate search directions $p_1,\ldots,p_n$ (so that $P^T AP$ is diagonal, i.e., (P,P) is an A-conjugate pair, where $P = [p_1,\ldots,p_n]$) and generating from a given y_0 the same sequence of iterates as the usual conjugate directions algorithm for symmetric A. In this sense the methods which concern us here are generalized conjugate directions algorithms. However, the conventional conjugate directions algorithms as well as the generalized ones that Stewart (1972) discussed (except for the bidiagonalization method of Golub and Kahan) use a conjugating algorithm to determine the search directions p_k; given a vector $\tilde{p}_k$ linearly independent of the preceding search directions $p_1,\ldots,p_{k-1}$, the conjugating algorithm adds multiples of $p_1,\ldots,p_{k-1}$ to $\tilde{p}_k$ to obtain a search direction p_k that is A-conjugate to the preceding conjugating directions $q_1,\ldots,q_{k-1}$ (i.e., $q_j^T Ap_k = p_k A^T q_j = 0$ for $1 \le j < k-1$). In contrast to this approach and with the exception of Craig's method (discussed in §2.5), the methods we consider below maintain a basis $u_k^{(k)},\dots,u_n^{(k)}$ for the orthogonal complement Span $\{A^Tq_1,\dots,A^Tq_{k-1}\}^\perp$ of $A^Tq_1,\dots,A^Tq_{k-1}$ and choose p_k as a linear combination of $u_k^{(k)},\dots,u_n^{(k)}$. In this respect the methods discussed here differ markedly from the conventional conjugate directions algorithms. While this distinction leads to more overhead when these methods are applied to a linear f or a quadratic ϕ, it should also lead to search directions which are more truly A-conjugate to the conjugating directions when inexact arithmetic is used.

This thesis is organized as follows. Section 1.2 introduces some notation and briefly reviews some background material. The appendix (§7) contains a more complete list of the special notation used. Section 2 discusses finding a zero of a continuously differentiable f: $\mathbb{R}^n \to \mathbb{R}^n$ by generalizations of Brown's method; in §2.1 we state the generalized algorithm and obtain a lower bound on its local rate of convergence; in §2.2 we examine Brown's method in detail and show how to implement it in such a way that the housekeeping operations require only $n^2/4 + O(n)$ storage locations and $O(n^3)$ arithmetic operations per major iteration, rather than the $n^2/2 + O(n)$ storage locations and $O(n^4)$ arithmetic operations required by Brown's implementation; in §2.3 we show how to use variable but finite precision arithmetic in Brown's method without disturbing the rate of convergence; we discuss Brent's variation of Brown's method in §2.4; in §2.5 we derive Craig's method for solving a system of linear equations; and in §2.6 we discuss some modifications to the generalized algorithm which may increase its range of applicability or improve its efficiency. Section 3 shows how to extend the generalized Brown's method of §2 to a descent algorithm for solving unconstrained minimization problems; in §3.1 we discuss how to choose the search directions p_k to insure, at least in the quadratic case, that the objective function decreases along the entire step in direction p_k; in §3.2 we examine the local form of the proposed minimization algorithms and establish its rate of convergence; in §3.3 we add various tests to the local algorithm of §3.2 and prove that the resulting subroutine may be combined with an appropriate calling program to yield a locally, rapidly convergent algorithm; we show in §3.4 how to construct a calling program which, when combined with the subroutine of §3.3, results in a globally and rapidly convergent descent algorithm (under the appropriate conditions); we specify more fully some particular versions of this algorithm in §3.5 and discuss the relationship between one of these and certain generalized conjugate gradient algorithms; and in §3.6 we discuss some preliminary computational experience and suggest certain modifications that may improve performance. Finally, §4 indicates how the approaches discussed in §2 and §3 extend to the constrained minimization problem, while §5 contains some concluding remarks.

1.2 Notation and Background

The appendix (§7) contains a list of most of the special notation used in this thesis. While we define some of our less standard notation the first time it is used, we mention a few items here at the outset before reviewing some background material.

The set of real numbers $(-\infty,\infty)$ will be denoted by $\mathbb{R}$, while $\mathbb{R}^n$ denotes Euclidean n-space (with the usual topology). We let e_j denote the j-th standard unit vactor $e_j = (0,\ldots,0,1,0,\ldots,0)^T$ of $\mathbb{R}^k$, where k will be clear from context; sometimes $e_i \in \mathbb{R}^k$ and $e_j \in \mathbb{R}^m$ may occur in the same formula with $k \neq m$, but this should cause no confusion. Usually we will use the 2-norm $\|x\| \equiv \|x\|_2 \equiv \left(\sum_{j=1}^{k} (e_j^T x)^2\right)^{1/2}$ of $x \in \mathbb{R}^k$, but sometimes we refer to the max-norm $\|x\|_\infty \equiv \max\{|e_j^T x| : 1 \le j \le k\}$. Given $\rho > 0$, i.e. $\rho \in (0,\infty]$, and $x \in \mathbb{R}^n$, we let $B(x,\rho) = \{y \in \mathbb{R}^n : \|x-y\| < \rho\}$ and $\bar{B}(x,\rho) = \{y \in \mathbb{R}^n : \|x-y\| \le \rho\}$. By $\mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ we mean the set of $m \times n$ matrices, and $\mathcal{L}(\mathbb{R}^n) \equiv \mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$. Finally, LHS and RHS denote the left- and right-hand sides, respectively, of the formula referenced. For example, LHS(4) means the left-hand side of formula (4) of the current subsection, while RHS(2.1.7) refers to the right-hand side of formula (7) of §2.1.

Rates of Convergence

Let $x_1, x_2, \ldots$ be a sequence of points in $\mathbb{R}^n$ converging to x^*. If there exists a sequence $c_1, c_2, \ldots$ of real numbers such that $\lim_{i\to\infty} c_i = 0$ and $\|x_{i+1}-x^*\| \le c_i\|x_i-x^*\|$ for all $i \ge 1$, then $x_1, x_2, \ldots$ are said to converge Q-superlinearly to x^*. We may define respectively the Q and R orders $\mathcal{O}_Q(x_1,x_2,\ldots)$ and $\mathcal{O}_R(x_1,x_2,\ldots)$ of convergence of this sequence by

$$\mathcal{O}_Q(x_1,x_2,\ldots) \equiv \sup\{\mu \in [0,\infty) \mid \text{there exists } c = c(\mu) \in \mathbb{R} \text{ such that } \|x_{i+1}-x^*\| \le c\|x_i-x^*\|^{\mu} \text{ for } i = 1,2,\ldots\} \text{ and}$$

$$\mathcal{O}_R(x_1,x_2,\ldots) \equiv \sup\{\mu \in [0,\infty) \mid (\lim_{i\to\infty} \|x_i-x^*\|^{1/(\mu^i)} = 0 \text{ and } \mu > 0) \text{ or } \mu = 0\}.$$

Dennis (in work not yet published) has shown that

$$\mathcal{O}_R(x_1,x_2,\ldots) = \sup\{\mathcal{O}_Q(t_1,t_2,\ldots) : \|x_i-x^*\| \le t_i \text{ and } \lim_{i\to\infty} t_i = 0\};$$

thus $\sigma_Q(x_1,x_2,\ldots) \le \sigma_R(x_1,x_2,\ldots)$.

The sequence $x_1,x_2,\ldots$ is said to converge Q-quadratically to x^* if it converges to x^* and there exists $c \in \mathbb{R}$ such that

$$\|x_{i+1}-x^*\| \le c\|x_i-x^*\|^2 \quad \text{for } i = 1,2,\ldots .$$

(Note that Q-quadratic convergence is slightly stronger than Q convergence of order 2; for example, if $x_1 = 1/2$ and $x_{i+1} = [\ell n(i+1)](x_i)^2$, then the sequence $x_1,x_2,\ldots$ converges to 0 with Q order 2 but does not converge Q-quadratically.) For more discussion about measuring rates of convergence, see Chapter 9 of (Ortega and Rheinboldt, 1970).

Singular Values

Assume $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ with $m \ge n$. A may be expressed as the product

$$A = Q^T \begin{pmatrix} \mathrm{diag}(d_1,\ldots,d_n) \\ \cdots\cdots\cdots\cdots \\ 0_{(m-n)\times n} \end{pmatrix} P,$$

where $P \in \mathcal{L}(\mathbb{R}^n)$ and $Q \in \mathcal{L}(\mathbb{R}^m)$ are orthogonal matrices ($P^{-1} = P^T$, $Q^{-1} = Q^T$) and $d_1,\ldots,d_n \ge 0$ are the singular values of A, i.e., the square roots of the eigenvalues of A^TA (see (Forsythe and Moler, 1967, p.10)); this factorization of A is called the singular values decomposition of A. Let the singular values be ordered so that $d_1 \ge d_2 \ge \ldots \ge d_n \ge 0$; then $d_n\|x\| \le \|Ax\| \le d_1\|x\|$ for any $x \in \mathbb{R}^n$, so $\|A\| = d_1$. If A has full column rank, so that $d_n > 0$, then we define the (2-norm) condition number $\chi(A) = \chi(A^T)$ to be $\chi(A) \equiv d_1/d_n$, so that $\|Ax\| \ge d_n\|x\| = \|A\|\|x\|/\chi(A)$ for all $x \in \mathbb{R}^n$.

If also $m = n$, then $A^{-1} = P^T \mathrm{diag}(d_1^{-1},\ldots,d_n^{-1})Q$ and $\|A^{-1}\| = d_n^{-1}$, so $\chi(A) = \|A\|\|A^{-1}\|$.

Orthogonal Projections

Often below we will refer to the orthogonal projection of a vector onto the column space of a certain matrix or onto the orthogonal complement thereof. Thus it will be convenient to use the notation of pseudoinverses. If, as above, $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ with $m \geq n$ has the singular values decomposition $A = Q^T \begin{pmatrix} \mathrm{diag}(d_1,\ldots,d_n) \\ \cdots\cdots\cdot \\ 0_{(m-n)\times n} \end{pmatrix} P$, then we define $d_i^+ \equiv \begin{cases} 1/d_i & \text{if } d_i \neq 0 \\ 0 & \text{if } d_i = 0 \end{cases}$

and $A^+ \equiv P^T\left(\mathrm{diag}(d_1^+,\ldots,d_n^+) \vdots 0_{n\times(m-n)}\right)Q$ and call A^+ the "pseudoinverse" of A. If $y \in \mathbb{R}^m$, then AA^+y is the orthogonal projection of y onto the column space of A, while $(I-AA^+)y$ is the orthogonal projection of A onto the orthogonal complement of the column space of A. If A has full column rank, then $A^+ = (A^TA)^{-1}A^T$; in particular, if the columns of A are orthonormal, then $A^TA = I$ and $A^+ = A^T$. Sometimes we will explicitly compute A^Ty and then talk about $AA^+y = (A^TA)^{-1}(A^Ty)$.

Floating-point Arithmetic

We will usually assume exact arithmetic; however, we will state our results in such a way that it will be relatively easy to extend them to finite but variable precision arithmetic. As an illustration, we will make such an extension for Brown's method in §2.3. When we deal with finite precision arithmetic, we will assume that floating-point operations are performed with maximum relative error bounded by a specified constant λ, called the "unit roundoff." Thus if $*$ denotes an arithmetic operation (addition, subtraction, multiplication, or division), if $c, d \in \mathbb{R}$ are floating-point numbers, and if $fl(c * d)$ denotes the result of computing $c * d$ in floating-point arithmetic, then $fl(c * d) = (c * d)(1 + \lambda_1)$, where $|\lambda_1| \leq \lambda$. See (Forsythe and Moler, 1967, §20) for a more detailed discussion.

Algorithms

We will state algorithms (some of which computation theorists would call "procedures," because under certain conditions they may not halt, i.e. may loop forever) and algorithmically defined functions in an informal language similar to ALGOL 60 or PL/I. In particular, the body of an iterative DO loop of the form

```
DO i = j TO k;
        body
        END;
```

will be executed with i having the successive values $j, j+1, \ldots, k$, provided $j \leq k$; if $j > k$, then i is assigned the value j and the body is not executed at all; and if $j \leq k$, then $i = k+1$ after the loop has been executed.

The assignment operator will be denoted by "$\leftarrow$" or "let ... =", e.g. "$x \leftarrow y$" or "let $x = y$".

Finite Difference Operators

Assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $h, h_1, h_2 \in (0, \infty)$, and $u, v \in \mathbb{R}^n$. In subsequent sections we will often have occasion to approximate $\frac{\partial\phi}{\partial u}(x) \equiv \phi'(x)u$ and $\frac{\partial^2\phi}{\partial u \partial v}(x) \equiv v^T\phi''(x)u$ by finite differences. Thus it is convenient to define the following finite difference operators:

(1) $\dfrac{\Delta\phi}{\Delta u}(x,h) \equiv \dfrac{\phi(x+hu)-\phi(x)}{h}$, the forward difference operator;

(2) $\dfrac{\check{\Delta}\phi}{\Delta u}(x,h) \equiv \dfrac{\phi(x)-\phi(x-hu)}{h} = \dfrac{-\Delta\phi}{\Delta(-u)}(x,h)$, the backward difference operator;

(3) $\dfrac{\tilde{\Delta}\phi}{\Delta u}(x,h) \equiv \dfrac{\phi(x+hu)-\phi(x-hu)}{2h}$, the central difference operator;

(4) $\dfrac{\Delta^2\phi}{\Delta u\Delta v}(x,h_1,h_2) \equiv \dfrac{\phi(x+h_1u+h_2v)-\phi(x+h_1u)-\phi(x+h_2v)+\phi(x)}{h_1h_2}$;

(5) $\dfrac{\Delta^2\phi}{\Delta u\Delta v}(x,h) \equiv \dfrac{\phi(x+hu+hv)-\phi(x+hu)-\phi(x+hv)+\phi(x)}{h^2} = \dfrac{\Delta^2\phi}{\Delta u\Delta v}(x,h,h)$;

(6) $\dfrac{\tilde{\Delta}^2\phi}{\Delta u^2}(x,h) \equiv \dfrac{\phi(x+hu)-2\phi(x)+\phi(x-hu)}{h^2}$;

(7) $\dfrac{\hat{\tilde{\Delta}}\phi}{\Delta u}(x,h) \equiv \dfrac{4\phi(x+hu)-3\phi(x)-\phi(x+2hu)}{2h}$;

(8) $\dfrac{\check{\tilde{\Delta}}\phi}{\Delta u}(x,h) \equiv \dfrac{-4\phi(x-hu)+3\phi(x)+\phi(x-2hu)}{2h} = \dfrac{-\hat{\tilde{\Delta}}\phi}{\Delta(-u)}(x,h)$;

(9) $\dfrac{\hat{\Delta}^2\phi}{\Delta u^2}(x,h) \equiv \dfrac{\phi(x)-2\phi(x+hu)+\phi(x+2hu)}{h^2} = \dfrac{\tilde{\Delta}^2\phi}{\Delta u^2}(x+hu,u)$;

(10) $\dfrac{\check{\Delta}^2\phi}{\Delta u^2}(x,h) \equiv \dfrac{\phi(x)-2\phi(x-hu)+\phi(x-2hu)}{h^2} = \dfrac{\hat{\Delta}^2\phi}{\Delta u^2}(x-2hu,u) = \dfrac{\tilde{\Delta}^2}{\Delta u^2}(x-hu,u)$.

In most of the algorithms given below we have in mind using $\dfrac{\Delta\phi}{\Delta u}$ or $\dfrac{\tilde{\Delta}\phi}{\Delta u}$, $\dfrac{\Delta^2\phi}{\Delta u\Delta v}(x,h)$ for $u \neq v$, and $\dfrac{\tilde{\Delta}^2\phi}{\Delta u^2}$; in practice, when we have only finite precision arithmetic of limited accuracy, we might use $\frac{\Delta^2\phi(x,h_1,h_2)}{\Delta u\Delta v}$ instead of $\frac{\Delta^2\phi}{\Delta u\Delta v}(x,h)$, so that we could better balance the roundoff and truncation errors. In constrained problems we might be obliged by the domain of ϕ to use $\frac{\check{\Delta}\phi}{\Delta u}$, $\frac{\tilde{\hat{\Delta}}\phi}{\Delta u}$, $\frac{\tilde{\check{\Delta}}\phi}{\Delta u}$, $\frac{\hat{\Delta}^2\phi}{\Delta u^2}$, and/or $\frac{\check{\Delta}^2\phi}{\Delta u^2}$. Of course, we could devise finite difference operators with a higher-order truncation error (which would allow the use of larger values of h to achieve the same accuracy and might thus be desirable when very accurate approximate derivatives are needed but arithmetic of only limited precision is available), but these would cost extra function evaluations to compute.

In later sections we will need some of the estimates given by the following

(11) Lemma: Let $x,u,v \in \mathbb{R}^n$, $h,h_1,h_2 \in (0,\infty)$, $\mu \in [0,1]$, $\mathcal{N} \subset \mathbb{R}^n$ be open, and $\phi\colon \mathcal{N} \to \mathbb{R}$. If ϕ is Frechet differentiable on $\mathcal{N}$ with

(12) $$\|\phi'(y) - \phi'(z)\| \leq \kappa_1\|y-z\|^\mu$$

for all $y,z \in \mathcal{N}$ and $\mathcal{N}$ contains the line segment connecting x and $x \pm hu$, then

(13) $$\left|\frac{\Delta\phi}{\Delta u}(x,h) - \frac{\partial\phi(x)}{\partial u}\right| \leq \kappa_1 h^\mu\|u\|^{1+\mu}$$

and

(14) $$\left|\frac{\check{\Delta}\phi}{\Delta u}(x,h) - \frac{\partial\phi(x)}{\partial u}\right| \leq \kappa_1 h^\mu\|u\|^{1+\mu}$$

Furthermore, if ϕ is twice Frechet differentiable on $\mathcal{N}$ with

(15) $$\|\phi''(y) - \phi''(z)\| \leq \kappa\|y-z\|^\mu$$

for all $y,z \in \mathcal{N}$ and $\mathcal{N}$ contains the line segments connecting x with $x+h_1u$, $x+h_2v$, $x \pm hu$, $x+hv$, and $x \pm 2hu$, then

$$(16)\qquad \left|\frac{\tilde{\Delta}\emptyset}{\Delta u}(x,h) - \frac{\partial\emptyset}{\partial u}(x)\right| \le \frac{h^{1+\mu}}{2^{2-\mu}}\|u\|^{2+\mu},$$

$$(17)\qquad \left|\frac{\Delta^2\emptyset}{\Delta u\Delta v}(x,h_1,h_2) - \frac{\partial^2\emptyset}{\partial u\partial v}(x)\right| \le \frac{\kappa}{2}(h_1\|u\|+h_2\|v\|)^{\mu}\left(\sqrt{\frac{h_1}{h_2}}\|u\|+\sqrt{\frac{h_2}{h_1}}\|v\|\right)^2,$$

$$(18)\qquad \left|\frac{\Delta^2\emptyset}{\Delta u\Delta v}(x,h) - \frac{\partial^2\emptyset}{\partial u\partial v}(x)\right| \le \frac{h^{\mu}\kappa}{2}(\|u\|+\|v\|)^{2+\mu},$$

$$(19)\qquad \left|\frac{\tilde{\Delta}^2\emptyset}{\Delta u^2}(x,h) - \frac{\partial^2\emptyset}{\partial u^2}(x)\right| \le h^{\mu}\kappa\|u\|^{2+\mu},$$

$$(20)\qquad \left|\frac{\hat{\tilde{\Delta}}\emptyset}{\Delta u}(x,h) - \frac{\partial\emptyset}{\partial u}(x)\right| \le 2^{\mu}h^{1+\mu}\kappa\|u\|^{2+\mu},$$

$$(21)\qquad \left|\frac{\check{\tilde{\Delta}}\emptyset}{\Delta u}(x,h) - \frac{\partial\emptyset}{\partial u}(x)\right| \le 2^{\mu}h^{1+\mu}\kappa\|u\|^{2+\mu},$$

$$(22)\qquad \left|\frac{\hat{\Delta}^2\emptyset}{\Delta u^2}(x,h) - \frac{\partial^2\emptyset}{\partial u^2}(x)\right| \le 2h^{\mu}\kappa\|u\|^{2+\mu},$$

and

$$(23)\qquad \left|\frac{\check{\Delta}^2\emptyset}{\Delta u^2}(x,h) - \frac{\partial^2\emptyset}{\partial u^2}(x)\right| \le 2h^{\mu}\kappa\|u\|^{2+\mu}.$$

Proof: We have (13) <=> (14), (17) => (18), (20) <=> (21), and (22) <=> (23). Moreover, since $\frac{\hat{\Delta}^2\emptyset}{\Delta u^2}(x,h) = \frac{\tilde{\Delta}^2\emptyset}{\Delta u^2}(x+hu,h)$, we see that (19) and (15) imply (22). Thus it suffices to show that (13), (16), (17), (19), and (20) are valid.

By the mean value theorem (MVT) there exists $\theta_1 \in (0,1)$ such that $\frac{\Delta\phi(x,h)}{\Delta u} \equiv \frac{\phi(x+hu)-\phi(x)}{h} = \phi'(x+\theta_1 hu)u$, whence (13) follows from (12).

By Taylor's theorem there exist $\theta_2,\theta_3 \in (0,1)$ such that

$$\frac{\tilde{\Delta}\phi}{\Delta u}(x,h) \equiv \frac{\phi(x+hu)-\phi(x-hu)}{2h} = \phi'(x)u + \frac{h}{4}u^T[\phi''(x+\theta_2 hu) - \phi''(x-\theta_3 hu)]u,$$

which, since $\frac{\partial\phi}{\partial u}(x) = \phi'(x)u$ and $\|(x+\theta_2 hu) - (x-\theta_3 hu)\| \le 2h\|u\|$, combines with (15) to imply (16).

Again by Taylor's theorem there exist $\theta_4,\theta_5,\theta_6 \in (0,1)$ such that

$$\frac{\Delta^2\phi}{\Delta u\Delta v}(x,h_1,h_2) \equiv \frac{\phi(x+h_1u+h_2v) - \phi(x+h_1u) - \phi(x+h_2v) + \phi(x)}{h_1h_2}$$

$$= \frac{1}{h_1h_2}\Big[\Big(\phi(x)+\phi'(x)(h_1u+h_2v) + \tfrac{1}{2}(h_1u+h_2v)^T\phi''(x+\theta_4[h_1u+h_2v])(h_1u+h_2v)\Big)$$

$$- \Big(\phi(x) + h_1\phi'(x)u + \tfrac{1}{2}h_1^2u^T\phi''(x+\theta_5h_1u)u\Big)$$

$$- \Big(\phi(x) + h_2\phi'(x)v + \tfrac{1}{2}h_2^2v^T\phi''(x+\theta_6h_2v)v\Big) + \phi(x)\Big]$$

$$= u^T\phi''(x+\theta_4[h_1u+h_2v])v + \frac{h_1}{2h_2}u^T[\phi''(x+\theta_4[h_1u+h_2v]) - \phi''(x+\theta_5h_1u)]u$$

(24)

$$+ \frac{h_2}{2h_1}v^T[\phi''(x+\theta_4[h_1u+h_2v]) - \phi''(x+\theta_6h_2v)]v,$$

which, together with (15), implies (17), since

$$\Big\|u^T\phi''(x+\theta_4[h_1u+h_2v])v - \frac{\partial^2\phi}{\partial u\partial v}(x)\Big\| = \|u^T[\phi''(x+\theta_4[h_1u+h_2v]) - \phi''(x)]v\|$$

$$\le \kappa(h_1\|u\| + h_2\|v\|)^{\mu}\|u\|\|v\|$$

and similar estimates hold for the last two terms of (24).

The proof of (19) is similar to (though simpler than) the proof of (17), and the proof of (20) is like that of (16). ∎

In practice, when several directional derivatives are approximated by finite differences at the same point, the differencing step length h is often varied with the differencing direction. To avoid further clutter below, we shall assume that a single differencing step length is used in such situations. However, the results obtained below remain unchanged if varying step lengths are used in these situations, so long as all lengths used lie in the interval specified for the single differencing step.

2. Solving Systems of Equations by Brown's Method and Generalizations Thereof

Assume throughout this section that $\underline{f}: \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable, i.e., $\underline{f} \in C^1(\mathbb{R}^n)$, and $x^* \in \mathbb{R}^n$ (with $n > 1$) is such that $\underline{f}(x^*) = 0$ and $\underline{f}'(x^*)$ is non-singular. We will often refer to such a point x^* as a nonsingular zero of f. In this section we consider using generalizations of Brown's method to produce a sequence $x_1, x_2, \ldots$ of points in $\mathbb{R}^n$, starting from a given $x_0 \in \mathbb{R}^n$, which converges to x^* whenever $\|x_0 - x^*\|$ is sufficiently small.

Starting from the major iterate x_{i-1}, the methods under consideration determine the next major iterate x_i by constructing a sequence of minor iterates $y_0, y_1, \ldots, y_n$ with $y_0 = x_{i-1}$ and then setting $x_i = y_n$. During this i-th major iteration, the methods rotate the range space, in that they choose an orthogonal matrix Q and work with $f \equiv Q^T\underline{f}$: as described in §1.1, they construct linear approximations $\bar{F}_k$ to the components f_k of f and attempt to choose y_k so that $\bar{F}_1(y_k) = \bar{F}_2(y_k) = \ldots = \bar{F}_k(y_k) = 0$. Now if $Q = [q_1, \ldots, q_n]$, so that q_k is the k-th column of Q, then $f_k = q_k^T\underline{f}$. Thus it is not necessary to know q_k until the time has come to construct $\bar{F}_k$, so Q may be generated dynamically during the major iteration. While Brown's method and Brent's variant thereof always choose Q = I, Craig's method for solving linear equations (discussed in §2.5) and the minimization algorithms proposed in §3 and §4 do construct Q "on the fly," i.e. one column at a time during the course of each major iteration. Thus these methods attempt to determine minor iterate y_k in such a way that $\underline{f}(y_k)$ vanishes in the directions $q_1, \ldots, q_k$, i.e., that $q_j^T\underline{f}(y_k) = f_j(y_k) = 0$ for $1 \le j \le k$.

We assume that $\underline{f}$ is defined on all of $\mathbb{R}^n$. While the results presented here remain valid of $\underline{f}: \mathcal{N} \to \mathbb{R}^n$, where $\mathcal{N}$ is an open subset of $\mathbb{R}^n$, and we somehow guarantee that the methods considered attempt to evaluate $\underline{f}$ only at points in $\mathcal{N}$, it simplifies the discussion and statement of results somewhat if $\mathbb{R}^n$ is the domain of f, which, of course, is implied by $\underline{f} \in C^1(\mathbb{R}^n)$.

We have in mind that finite difference operators, such as those defined in §1.2, will be used to compute approximate partial derivatives. Rather than specify which operator to use, we shall simply require that approximations be made, the error in which is bounded by a constant times an error factor ξ. To allow for the use of finite-precision arithmetic, we similarly allow an error bounded by a multiple of ξ in the results of certain arithmetic calculations.

2.1 The Generalized Algorithm and Its Convergence

In this subsection we state the generalization of Brown's method with which we shall subsequently work and prove a local convergence theorem. Note that P_k, defined below, is a projection matrix and not a matrix whose columns are search directions p_k.

Like Brown's method as described in §1.1, the generalized algorithms considered here construct a linear approximation $\bar{F}_k$ to the component function f_k of $f \equiv Q^T\underline{f}$ by implicitly finding an approximation $\bar{w}_k$ to $\nabla f_k(y_{k-1})$ and setting $\bar{F}_k(y) \equiv f_k(y_{k-1}) + \bar{w}_k^T(y-y_{k-1})$. Starting from $y_0 = x_{i-1}$, the current major iterate, these methods attempt to determine $y_1,\ldots,y_n$ so that $\bar{F}_1(y_k) = \ldots = \bar{F}_k(y_k) = 0$, $1 \le k \le n$. (The reason why we say "attempt to determine" rather than "determine" will be explained presently.) Having found y_{k-1} and $\bar{F}_1,\ldots,\bar{F}_{k-1}$, the methods attempt to choose

y_k in the form $y_k = y_{k-1} - \lambda_k p_k$, where $\bar{w}_j^T p_k = 0$ for $1 \le j \le k-1$, so that $\bar{F}_j(y_{k-1}) = \bar{F}_j(y_k)$, $1 \le j \le k-1$.

Let $\bar{W}_j = [\bar{w}_1, \ldots, \bar{w}_j]$. Then p_k lies in the orthogonal complement of the column space of $\bar{W}_{k-1}$, i.e. $p_k \in \mathrm{Span}\{\bar{w}_1, \ldots, \bar{w}_{k-1}\}^{\perp}$, so $\bar{W}_{k-1}^{+} p_k = 0$ and $p_k = (I - \bar{W}_{k-1}\bar{W}_{k-1}^{+})p_k$. Let $P_j = I - \bar{W}_j\bar{W}_j^{+}$ (with $P_0 = I$), so that P_j projects orthogonally onto $\mathrm{Span}\{\bar{w}_1, \ldots, \bar{w}_j\}^{\perp}$, and let $w_j = P_{j-1}\bar{w}_j$. By induction we see that $\mathrm{Span}\{\bar{w}_1, \ldots, \bar{w}_j\} = \mathrm{Span}\{w_1, \ldots, w_j\}$. Hence if

(1a) $$W_j \equiv [w_1, \ldots, w_j]$$

then $W_j W_j^{+} = \bar{W}_j \bar{W}_j^{+}$, so

(1b) $$P_j \equiv I - W_j W_j^{+},$$

and in order that $\bar{F}_j(y_{k-1} - \lambda p_k) = \bar{F}_j(y_{k-1})$ for $j < k$, it is necessary and sufficient that $\lambda p_k = \lambda P_{k-1} p_k$. Moreover, it suffices to replace $\bar{F}_j$ by the F_j defined by

(2) $$F_j(y) \equiv f_j(y_{j-1}) + w_j^T(y - y_{j-1}),$$

since $\bar{F}_1(y_k) = \ldots = \bar{F}_k(y_k) = 0$ if $F_1(y_k) = \ldots = F_k(y_k) = 0$, $1 \le k \le n$, as long as $p_k = P_{k-1}p_k$. The algorithms under consideration therefore find approximations w_k to $P_{k-1}\nabla f_k(y_{k-1})$, with P_{k-1} as in (1).

The methods discussed in §3 and §4 seek to minimize $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and so have $f = \nabla\phi$. Some of these methods construct <u>two</u> linearizations per minor iteration: first they attempt to find $p_k \in \mathrm{Span}\{w_1, \ldots, w_{k-1}\}^{\perp}$, i.e.

p_k with $p_k = P_{k-1}p_k$, find an approximation π_k to $f_k'(y_{k-1})p_k$ and thus construct an approximation $\psi(\lambda) \equiv f_k(y_{k-1}) - \lambda\pi_k$ to $f_k(y_{k-1}-\lambda p_k)$. After choosing y_k, these methods implicitly construct F_k as defined in (2) by (implicitly) finding an approximation w_k to $P_{k-1}\nabla f_k(y_{k-1})$ or $P_{k-1}\nabla f_k(y_k)$. To choose y_k, they define $\lambda_k = f_k(y_{k-1})/\pi_k$, so that $\psi(\lambda_k) = 0 \approx f_k(y_{k-1}-\lambda_k p_k)$, and choose $y_k = y_{k-1} - \lambda_k p_k$ unless $\phi(y_{k-1}-\lambda_k p_k) > \phi(y_{k-1})$ (which, due to truncation and roundoff error, can happen if $\|\lambda_k p_k\|$ is very small compared to $\|\nabla\phi(y_0)\|$), in which case $y_k = y_{k-1}$. Either way it can happen that $F_k(y_k)$ vanishes only approximately. This, together with the fact that we may be using finite precision arithmetic, explains why we stated that the methods considered here <u>attempt to</u> determine y_k with $F_1(y_k) = \bar{F}_1(y_k) = \ldots = F_k(y_k) = 0$. This also explains why we allow y_k to be selected from a certain neighborhood rather than chosen to be one particular point in the generalized algorithm which we now discuss.

Recall that $\frac{\partial f_k}{\partial p_k}(y_{k-1}) \equiv f_k'(y_{k-1})p_k = p_k^T\nabla f_k(y_{k-1})$ and consider the following

(3) <u>Algorithm</u> [The inner loop of the generalized Brown's method];

(3a) Let $\underline{f} \in C^1(\mathbb{R}^n)$, $y_0 \in \mathbb{R}^n$, $\xi \in [0,\infty)$, and the constants $\delta, \epsilon_1, \epsilon_2, \epsilon_3, \eta$ be given;

DO $k = 1$ TO n;

(3b) $\begin{cases} \text{Choose } q_k \in \text{Span}\{q_1,\ldots,q_{k-1}\}^{\perp} \text{ with } \|q_k\| = 1 \\ \text{(so that } q_1,\ldots,q_k \text{ are orthonormal) and let } f_k \equiv q_k^T\underline{f}\,; \end{cases}$

Let $W_{k-1} = [w_1,\ldots,w_{k-1}]$ and $P_{k-1} = I - W_{k-1}W_{k-1}^+$, as in (1);

(3c) Choose $p_k \neq 0$ with

(3c(i)) $\|p_k - P_{k-1}p_k\| \le \xi\delta\|p_k\|$ and

(3c(ii)) $\left|\dfrac{\partial f_k}{\partial p_k}(y_{k-1})\right| \ge \eta\|P_{k-1}\nabla f_k(y_{k-1})\|\|p_k\|$.

(3d) Find π_k with $|\pi_k - \dfrac{\partial f_k}{\partial p_k}(y_{k-1})| \le \epsilon_1\xi\|p_k\|$;

(3e) Choose $y_k \in \bar{B}\left(y_{k-1} - \left(\dfrac{f_k(y_{k-1})}{\pi_k}\right)p_k,\ \xi\epsilon_2\|\underline{f}(y_0)\|\right)$, i.e., choose y_k so that $\|y_k - y_{k-1} + \dfrac{f_k(y_{k-1})}{\pi_k}p_k\| \le \xi\epsilon_2\|\underline{f}(y_0)\|$;

(3f) Find w_k with $\|w_k - P_{k-1}\nabla f_k(y_{k-1})\| \le \epsilon_3\xi$;

END; END Algorithm (3);

To facilitate the analysis of the methods considered here when finite precision arithmetic is used, we do not specify $p_k \in \mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$, i.e. $p_k = P_{k-1}p_k$, but rather that p_k have only a small component in $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}$; condition (3c(i)) makes this requirement precise. As we shall soon see, conditions (3c(ii)) and (3d) will assure that π_k is bounded away from 0, at least when all the minor iterates $y_0,\ldots,y_{k-1}$ are close enough to x^*.

An example may help clarify how Algorithm (3) works. Let $\underline{f}(x) = Ax$, where $A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 0 \\ 2 & 1 & 1 \end{pmatrix}$, so that $f(x) = 0$ iff $x = x^* = 0$. Suppose $y_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

For simplicity, we will choose $q_k = e_k$, whence $Q = I$ and $f \equiv Q^T\underline{f} = \underline{f}$. We will do exact computations, i. e. $\xi = 0$.

Iteration 1: Arbitrarily choose $p_1 = e_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}$. Since $f_1(x) = (1\ 0\ 1)x$, we have

$$\pi_1 = \frac{\partial f_1}{\partial p_1}(y_0) = f_1'(y_0)p_1 = (1\ 0\ 1)\begin{pmatrix}1\\0\\0\end{pmatrix} = 1.$$

Since $f_1(y_0) = 2$,

$$y_1 = y_0 - \frac{f_1(y_0)}{\pi_1}\, p_1 = \begin{pmatrix}1\\1\\1\end{pmatrix} - \frac{2}{1}\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}-1\\1\\1\end{pmatrix}.$$

Since $P_0 = I$,

$$w_1 = P_0\nabla f_1(y_0) = \begin{pmatrix}1\\0\\1\end{pmatrix}.$$

Iteration 2: Let $p_2 = e_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}$: since $w_1^Tp_2 = 0$, this choice satisfies (3c(i)). Noting that $f_2(x) = (1\ 2\ 0)x$, we obtain

$$\pi_2 = \frac{\partial f_2}{2p_2}(y_1) = (1\ 2\ 0)\begin{pmatrix}0\\1\\0\end{pmatrix} = 2.$$

Since $f_2(y_1) = 1$,

$$y_2 = y_1 - \frac{f_1(y_1)}{\pi_2} = \begin{pmatrix}-1\\1\\1\end{pmatrix} - \frac{1}{2}\begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}-1\\1/2\\1\end{pmatrix}.$$

Now

$$P_1 = I - [w_1][w_1]^+ = I - \frac{w_1w_1^T}{\|w_1\|^2}, \quad \text{and} \quad \nabla f_2(y_1) = \begin{pmatrix}1\\2\\0\end{pmatrix},$$

so

$$w_2 = P_1 \nabla f_2(y_1) = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \frac{1}{2} = \begin{pmatrix} 1/2 \\ 2 \\ -1/2 \end{pmatrix}.$$

Iteration 3: Let $p_3 = \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix}$: then $w_1^T p_3 = w_2^T p_3 = 0$, so (3c(i)) is satisfied. Since $f_3(x) = (2\ 1\ 1)x$,

$$\pi_3 = \frac{\partial f_3}{\partial p_3}(y_2) = (2\ 1\ 1) \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} = 1, \text{ and}$$

$$y_3 = y_2 - \frac{f_3(y_2)}{\pi_3} p_3 = \begin{pmatrix} -1 \\ 1/2 \\ 1 \end{pmatrix} - \frac{(-1/2)}{1} \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} = 0 = x^*.$$

Since w_3 and P_2 are not used, they need not be computed.

Notice that $f_j(y_k) = 0$ for $1 \leq j \leq k \leq 3$, as desired.

We next state and prove some lemmas leading to the result that if f' is continuous and satisfies a Lipschitz condition of order $\mu \in [0,1]$ near x^*, i.e. if

$$\|\underline{f}'(x) - \underline{f}'(y)\| \leq \kappa \|x-y\|^\mu \quad \forall x,y \in B(x^*, \rho_2)$$

for some $\rho_2 \in (0,\infty]$ and $\kappa \in [0,\infty)$, then there exists a neighborhood $B(x^*,\rho)$ of x^* and constants K_8, K_9 such that

$$\|y_n - x^*\| \leq (K_8 \xi + K_9 \kappa \|y_0 - x^*\|^\mu) \|y_0 - x^*\|$$

whenever $y_0 \in B(x^*,\rho)$. We will subsequently see how to arrange that $\xi = \mathcal{O}(\kappa\|y_0 - x^*\|^\mu)$, which leads to nonlinear equations solving algorithms having local convergence of Q-order at least $1+\mu$. The upcoming analysis will also let us estimate the rate of convergence when the same values of $p_1, \ldots, p_n$

and $\pi_1,\ldots,\pi_n$ are used in several major iterations (see §2.6).

First we obtain a bound on the step length $\|y_k-y_{k-1}\|$ taken in Algorithm (3) as well as a bound on $|f_k(y_k)|$:

(4) <u>Lemma</u>: Let $\mu \in [0,1]$, $\eta \in (0,1]$, $\rho_1 \in (0,\infty]$, and κ, δ, ϵ_1, ϵ_2, $\epsilon_3 \in [0,\infty)$ be fixed and assume $f \in C^1(\mathbb{R}^n)$. During the k-th iteration of Algorithm (3), if

(5) $$\eta\|P_{k-1}\nabla f_k(y_{k-1})\| > \epsilon_1\xi, \quad \text{then} \quad \pi_k \neq 0 \quad \text{and}$$

(6) $$\|y_k-y_{k-1}\| \leq \xi\epsilon_2\|\underline{f}(y_0)\| + \frac{|f_k(y_{k-1})|}{\eta\|P_{k-1}\nabla f_k(y_{k-1})\|-\epsilon_1\xi}.$$

Moreover, if y_{k-1}, $y_k \in B(x^*,\rho_1)$ and $\underline{f}'$ satisfies

(7) $$\|\underline{f}'(x)-\underline{f}'(y)\| \leq \kappa\|x-y\|^\mu \quad \forall\, x,y \in B(x^*,\rho_1) \quad \text{and}$$

(8) $$\|\underline{f}'(y)\| \leq K_1 \quad \forall\, y \in B(x^*,\rho_1),$$

then

(9) $$|f_k(y_k)| \leq K_1\epsilon_2\xi\|\underline{f}(y_0)\| + |f_k(y_{k-1})|\left[\frac{\kappa\|y_k-y_{k-1}\|^\mu+\epsilon_1\xi}{\eta\|P_{k-1}\nabla f_k(y_{k-1})\|-\epsilon_1\xi}\right].$$

<u>Proof</u>: We have

$$|\pi_k| \geq \left|\frac{\partial f_k}{\partial p_k}(y_{k-1})\right| - \left|\pi_k - \frac{\partial f_k}{\partial p_k}(y_{k-1})\right|$$

(10) $$\geq \|p_k\|(\eta\|P_{k-1}\nabla f_k(y_{k-1})\| - \epsilon_1\xi) \quad \text{by (3c(ii)) and (3d)}$$

$$> 0 \quad \text{by (3c) and (5).}$$

Since (3e) implies $\|y_k - y_{k-1}\| \le \xi\epsilon_2\|\underline{f}(y_0)\| + |f_k(y_{k-1})|\frac{\|p_k\|}{|\pi_k|}$, (6) follows from (3e) and (10). Assume now that (7) and (8) also hold. Since $f_k \equiv q_k^T\underline{f}$ and $\|q_k\| = 1$, (7) and (8) hold with $\underline{f}'$ replaced by f_k'. By the MVT (mean value theorem) there exists $\theta \in (0,1)$ such that

$$\begin{aligned} f_k(y_k) &= f_k(y_{k-1}) + f_k'(y_{k-1}+\theta[y_k-y_{k-1}])(y_k-y_{k-1}) \\ &= f_k'(y_{k-1}+\theta[y_k-y_{k-1}])\left[y_k-y_{k-1}+\frac{f_k(y_{k-1})}{\pi_k}\,p_k\right] + \\ &\quad + \left[f_k'(y_{k-1})-f_k'(y_{k-1}+\theta[y_k-y_{k-1}])\right]\frac{f_k(y_{k-1})p_k}{\pi_k} + \\ &\quad + f_k(y_{k-1})\left[1-\frac{f_k'(y_{k-1})p_k}{\pi_k}\right]. \end{aligned}$$

From (3d) and (10) we have $\left|1-\frac{f_k'(y_{k-1})p_k}{\pi_k}\right| \le \frac{\epsilon_1\xi}{\eta\|P_{k-1}\nabla f_k(y_{k-1})\| - \epsilon_1\xi}$,

whence, using (8), (3e), (10), and the convexity of $B(x^*,\rho_1)$,

$$\begin{aligned} |f_k(y_k)| \le{}& K_1\epsilon_2\xi\|\underline{f}(y_0)\| + \kappa\theta^\mu\|y_k-y_{k-1}\|^\mu|f_k(y_{k-1})|/[\eta\|P_{k-1}\nabla f_k(y_{k-1})\| - \epsilon_1\xi] \\ &+ |f_k(y_{k-1})|\frac{\epsilon_1\xi}{\eta\|P_{k-1}\nabla f_k(y_{k-1})\|-\epsilon_1\xi}, \end{aligned}$$

from which (9) follows, since $\theta^\mu \le 1$. ∎

Our next immediate goal is to obtain a lower bound on $\|P_{k-1}\nabla f_k(y_{k-1})\|$ that is valid whenever $y_0,\ldots,y_{k-1}$ lie close enough to x^*; the following two lemmas accomplish this. In the next lemma we have in mind that $v_j = \nabla f_j(x^*)$ and $\bar{w}_j = w_j + W_{j-1}W_{j-1}^+\nabla f_j(y_{j-1})$, so that $w_j = P_{j-1}\bar{w}_j$ and

$\|w_j - P_{j-1}\nabla f_j(y_{j-1})\| = \|\bar{w}_j - \nabla f_j(y_{j-1})\|$, where W_{j-1} and P_{j-1} are given in (1). (We may regard Algorithm (3) as implicitly constructing the linear approximation $\bar{F}_k(y) \equiv f_k(y_{k-1}) + \bar{w}_k^T(y-y_{k-1})$ during the k-th iteration.)

(11) Lemma: Let $v_1,\ldots,v_n$, $\bar{w}_1,\ldots,\bar{w}_{k-1} \in \mathbb{R}^n$ and define $V_j = [v_1,\ldots,v_j]$, $\bar{W}_j = [\bar{w}_1,\ldots,\bar{w}_j]$. If V_n is nonsingular, so that its 2-condition number $\chi(V_n) \equiv \|V_n\|\|V_n^{-1}\|$ is finite, if $1 \leq k \leq n$, and if

(12) $$\|\bar{w}_j - v_j\| \leq \zeta\|V_n\| \quad \text{for } 1 \leq j \leq k-1,$$

then

(13) $$\|(I-\bar{W}_{k-1}\bar{W}_{k-1}^{+})v_k\| \geq \frac{1-\zeta\chi(V_n)\sqrt{k-1}}{\|V_n^{-1}\|} .$$

Proof: We may assume that $\zeta\chi(V_n)\sqrt{k-1} < 1$, since (13) holds trivially otherwise. Consider the matrix $\mathcal{U} = [\bar{w}_1,\ldots,\bar{w}_{k-1}, v_k,\ldots,v_n]$: we have $\mathcal{U} - V_n = [\bar{w}_1-v_1,\ldots,\bar{w}_{k-1}-v_{k-1}, 0,\ldots,0]$. Thus if $x = (x_1,\ldots,x_n)^T \in \mathbb{R}^n$ has $\|x\| \equiv \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} = 1$, then (12) implies

$$\|(\mathcal{U}-V_n)x\| \leq \zeta\|V_n\| \sum_{i=1}^{k-1} |x_i| \leq \zeta\|V_n\|\sqrt{k-1} ,$$

whence $\|\mathcal{U}-V_n\| \leq \zeta\|V_n\|\sqrt{k-1}$. Since $\|V_n^{-1}\|\|\mathcal{U}-V_n\| \leq \zeta\|V_n^{-1}\|\|V_n\|\sqrt{k-1} = \zeta\chi(V_n)\sqrt{k-1} < 1$, $\mathcal{U}$ is nonsingular with

(14) $$\|\mathcal{V}^{-1}\| \le \frac{\|V_n^{-1}\|}{1-\zeta\chi(V_n)\sqrt{k-1}} .$$

Now we can express $P_{k-1}v_k = (I-\bar{W}_{k-1}\bar{W}_{k-1}^{+})v_k$ as $P_{k-1}v_k = \hat{v}_k + \tilde{v}_k$, where $\tilde{v}_k = P_{k-1}v_k - \hat{v}_k$ and $\hat{v}_k$ is the orthogonal projection of v_k onto $\mathrm{Span}\{\bar{w}_1,\ldots,\bar{w}_{k-1}, v_{k+1},\ldots,v_n\}^{\perp}$, i.e. $\hat{v}_k = (I-\underline{\mathcal{V}}\underline{\mathcal{V}}^{+})v_k$, where $\underline{\mathcal{V}} = [\bar{w}_1,\ldots, \bar{w}_{k-1}, v_{k+1},\ldots,v_n]$. Note that $(I-\underline{\mathcal{V}}\underline{\mathcal{V}}^{+})v_k = (I-\underline{\mathcal{V}}\underline{\mathcal{V}}^{+})P_{k-1}v_k$, whence $\tilde{v}_k = P_{k-1}v_k - \hat{v}_k$ is orthogonal to $\hat{v}_k$. Since $\hat{v}_k^T\tilde{v}_k = 0$, $\|P_{k-1}v_k\| \ge \|\hat{v}_k\|$. But $\hat{v}_k^T$ is a positive multiple of $u_k^T = e_k^T\mathcal{V}^{-1}$, i.e., the k-th row of $\mathcal{V}^{-1}$. Since $\|u_k\| \le \|\mathcal{V}^{-1}\|$ and $u_k^T\hat{v}_k = u_k^Tv_k = 1$, we have $\|u_k\|\|\hat{v}_k\| \ge u_k^T\hat{v}_k = 1$, whence

$$\|(I-\bar{W}_{k-1}\bar{W}_{k-1}^{+})v_k\| = \|P_{k-1}v_k\| \ge \|\hat{v}_k\| \ge 1/\|u_k\| \ge 1/\|\mathcal{V}^{-1}\|,$$

which together with (14) implies (13). ∎

(15) Lemma: Let $\underline{f} \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ be such that $\underline{f}'(x^*)$ is nonsingular. For each $\theta \in (0,1)$ and $\epsilon_3 \in [0,\infty)$, there exist ξ_0, $\rho_1 \in (0,\infty)$ such that for any k with $1 \le k \le n$, if $w_1,\ldots,w_{k-1}$ are generated as in (3b) and (3f) with $y_0,\ldots,y_{k-1} \in B(x^*,\rho_1)$, $\xi \le \xi_0$, and P_j as in (1), then

(16) $$\|P_{k-1}\nabla f_k(y_{k-1})\| \ge \frac{\theta}{\|\underline{f}'(x^*)^{-1}\|} .$$

Proof: for $n = 1$ this follows from the continuity of $\underline{f}'$. For $n > 1$, fix k and for $j < k$ let

(17) $$\bar{w}_j = w_j + W_{j-1}W_{j-1}^{+}\nabla f_j(y_{j-1}) \quad \text{and} \quad \bar{W}_j = [\bar{w}_1,\ldots,\bar{w}_j].$$

As previously remarked, we have $\|\bar{w}_j-\nabla f_j(y_{j-1})\| = \|w_j-P_{j-1}\nabla f_j(y_{j-1})\|$ and $P_{k-1} \equiv I - W_{k-1}W_{k-1}^{+} = I - \bar{W}_{k-1}\bar{W}_{k-1}^{+}$. From (3f) we have

(18) $$\|\bar{w}_j-\nabla f_j(y_{j-1})\| \le \epsilon_3\xi.$$

Let $Q = [q_1, \ldots, q_n]$ be an orthogonal matrix, where $q_1,\ldots,q_k$ are as chosen in step (3b), and let $f \equiv Q^T\underline{f}$. Then $\|f'(x)^{\pm 1}\| = \|\underline{f}'(x)^{\pm 1}\|$ and $f \in C^1(\mathbb{R}^n)$, so for each $\hat{\kappa} > 0$ there exists $\rho_1 > 0$ such that

(19a) $$\|f'(y)-f'(x^*)\| \le \hat{\kappa} \quad \text{whenever} \quad y \in B(x^*,\rho_1).$$

Let $\xi_0,\hat{\kappa} \in (0,\infty)$ be so small that

(19b) $$\epsilon_3\xi_0 + \hat{\kappa} \le \frac{1-\theta}{2\|\underline{f}'(x^*)^{-1}\|\sqrt{n-1}},$$

and let $\rho_1 > 0$ be so small that (19a) holds. Setting $v_j = \nabla f_j(x^*)$, $1 \le j \le n$, we see from (18), (19a), and the inequality $\|f_j'(y)-f_j'(x^*)\| \le \|f'(y)-f'(x^*)\|$ that $\|\bar{w}_j-v_j\| \le \epsilon_3\xi_0 + \hat{\kappa} \le \dfrac{1-\theta}{2\|\underline{f}'(x^*)^{-1}\|\sqrt{n-1}} = \left(\dfrac{1-\theta}{2\chi(f'(x^*))\sqrt{n-1}}\right)\|f'(x^*)\|$ whenever (3f) holds with $\xi \le \xi_0$ and $y_0,\ldots,y_{k-1} \in B(x^*,\rho_1)$. Under these conditions, let $V_n = f'(x^*)$ and $\zeta = \dfrac{1-\theta}{2\chi(V_n)\sqrt{n-1}}$; then Lemma (11) implies $\|P_{k-1}\nabla f_k(x^*)\| \ge \dfrac{\theta+1}{2\|\underline{f}'(x^*)\|}$.

But (19) implies

$$\|P_{k-1}(\nabla f_k(y_{k-1})-\nabla f_k(x^*))\| \le \|f'(y_{k-1})-f'(x^*)\| \le \hat{\kappa} \le \frac{1-\theta}{2\|\underline{f}'(x^*)\|},$$

so $\|P_{k-1}\nabla f_k(y_{k-1})\| \ge \frac{(\theta+1)-(1-\theta)}{2\|\underline{f}'(x^*)\|}$, which is equivalent to (16). ∎

Note that if $\underline{f}$ satisfies the Lipschitz condition $\|\underline{f}'(x)-\underline{f}'(y)\| \le \kappa\|x-y\|^{\mu}$, where $\mu \in [0,1]$, then ξ_0, $\rho_1 > 0$ satisfy the conclusion of Lemma (15) if $\epsilon_3\xi_0 + \kappa\rho_1^{\mu} \le \frac{1-\theta}{\|\underline{f}'(x^*)^{-1}\|\sqrt{n-1}}$.

We now derive, in terms of $\|\underline{f}(y_0)\|$ and $\|y_0-x^*\|$, some useful bounds on such quantities as $\|y_k-y_{k-1}\|$, $\|y_k-x^*\|$, and $\|\underline{f}(y_k)\|$. To do so we shall use all the assumptions stated in §2.0, namely that $\underline{f} \in C^1(\mathbb{R}^n)$, $\underline{f}(x^*)=0$, and $\underline{f}'(x^*)$ is nonsingular. Since $\|\underline{f}(x)\| = \|Q^T\underline{f}(x)\|$ for any orthogonal Q, we state these results in terms of $f \equiv Q^T\underline{f}$.

(20) <u>Lemma</u>: Let the constants δ, ϵ_1, ϵ_2, ϵ_3, η mentioned in Algorithm (3) be fixed; assume $\underline{f} \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ with $\underline{f}(x^*) = 0$ and $\underline{f}'(x^*)$ non-singular. There exist ξ_0, ρ_1, and $K_i \in (0,\infty)$, $0 \le i \le 4$, such that if Algorithm (3) runs for k iterations ($1 \le k \le n$) with $\xi \le \xi_0$ and $y_0,\ldots,y_{k-1} \in B(x^*,\rho_1)$ and if $Q = [q_1,\ldots,q_n]$ is any orthogonal matrix such that q_j is selected in step (3b) during iteration j, $1 \le j \le k$, then $f \equiv Q^T\underline{f}$ satisfies

(21) $$\|x-x^*\| \le K_0\|f(x)\| \qquad \forall\ x \in B(x^*,\rho_1),$$

(22) $$\|f(x)\| \le K_1\|x-x^*\| \qquad \forall\ x \in B(x^*,\rho_1),$$

(23) $$\|y_k - y_{k-1}\| \le K_3^{k-1} K_4 \|f(y_0)\|,$$

and

(24) $$\|y_k - x^*\| \le \left(K_0 + (K_3^k - 1)\frac{K_4}{K_1K_2}\right)\|f(y_0)\| .$$

Moreover, (23) and (24) hold with

(25) $$K_3 = 1 + K_1K_2$$

and

(26) $$K_4 = \xi_0\epsilon_2 + K_0K_1K_2.$$

Proof: By the continuity of f' and nonsingularity of $f'(x^*)$ there exists $\rho_1 > 0$ so small that $\|f'(x) - f'(x^*)\| \le \dfrac{1}{2\|f'(x^*)^{-1}\|}$ for all $x \in B(x^*,\rho_1)$. Hence if $x \in B(x^*,\rho_1)$ then

$$\|f(x)\| = \|\int_0^1 f'(x^*+t[x-x^*])(x-x^*)dt\| \qquad (\text{since } f(x^*) = 0)$$

$$\ge \|f'(x^*)(x-x^*)\| - \int_0^1 \|f'(x^*+t[x-x^*]) - f'(x^*)\|\|x-x^*\|dt$$

$$\ge \frac{\|x-x^*\|}{\|f'(x^*)^{-1}\|} - \|x-x^*\| \int_0^1 \frac{dt}{2\|f'(x^*)\|}$$

$$= \frac{\|x-x^*\|}{2\|f'(x^*)^{-1}\|} ,$$

whence (21) holds with $K_0 = 2\|f'(x^*)^{-1}\|$. Similarly, (22) holds with

(27) $$K_1 = \sup\|f'(B(x^*,\rho_1))\|.$$

Assume now that Algorithm (3) runs with $y_0,\dots,y_{k-1} \in B(x^*,\rho_1)$ and $\xi \leq \xi_0$. By Lemma (15) we may assume that (16) holds with $\theta = 1/2$, i.e.

(28a) $$\|P_{j-1}\nabla f_j(y_{j-1})\| \geq \frac{1}{2\|f'(x^*)^{-1}\|}, \qquad 1 \leq j \leq k.$$

Moreover, we may assume ξ_0 to be so small that

(28b) $$\frac{\eta}{2\|f'(x^*)^{-1}\|} - \epsilon_1\xi_0 > 0.$$

Let

(29) $$K_2 = \left(\frac{\eta}{2\|f'(x^*)^{-1}\|} - \epsilon_1\xi_0\right)^{-1}.$$

From (6) and the inequality $|f_k(y_{k-1})| \leq \|f(y_{k-1})\|$ we have

(30) $$\|y_k-y_{k-1}\| \leq \xi\epsilon_2\|f(y_0)\| + K_2\|f(y_{k-1})\|.$$

Let K_3 and K_4 be given by (25) and (26). Note that (24) follows from (23) and (21). Note also that (21) and (22) imply $K_0K_1 > 1$. Thus (30) implies that (23) and hence (24) hold for k = 1. Assume inductively that

$$\|y_j-y_{j-1}\| \leq K_3^{j-1}K_4\|f(y_0)\| \qquad \text{for } 1 \leq j \leq k-1$$

with $k > 1$, $k \leq n$. Then, by (24),

$$\|y_{k-1}-x^*\| \leq \left(K_0 + (K_3^{k-1}-1)\frac{K_4}{K_1K_2}\right)\|f(y_0)\|,$$

so (30), (22), and (26) imply

$$\|y_k - y_{k-1}\| \le \|f(y_0)\|\xi_0\epsilon_2 + K_1K_2\|y_{k-1} - x^*\|$$

$$\le \|f(y_0)\|[\xi_0\epsilon_2 + K_0K_1K_2 + (K_3^{k-1} - 1)K_4]$$

$$= K_3^{k-1}K_4\|f(y_0)\|,$$

which completes the inductive step and therewith the proof. ■

Lemma (4) gives a bound on $|f_k(y_k)|$. The next lemma will enable us to bound $|f_\nu(y_k)|$ for $\nu < k$ less pessimistically than (22) and (24) would suggest.

(31) Lemma: With the assumptions of Lemma (20) and ξ_0, ρ_1, K_1, K_2 as in that lemma, if

(32) $$\|\underline{f}'(x) - \underline{f}'(y)\| \le \kappa\|x - y\|^\mu \qquad \forall\ x,y \in B(x^*,\rho_1)$$

for some $\mu \in [0,1]$ and $\kappa \ge 0$, then there exists $K_5 \ge 0$ such that

(33) $$|f_\nu(y_\nu)(y_k - y_{k-1})| \le K_1\xi\epsilon_2\|f(y_0)\| + K_2|f_k(y_{k-1})|(\kappa\|y_\nu - y_{\nu-1}\|^\mu + K_5\xi)$$

whenever $1 \le \nu < k \le n$ and Algorithm (3) runs with $y_0,\ldots,y_k \in B(x^*,\rho_1)$ and $\xi \le \xi_0$.

Proof: Recall (1). Since $\nu < k$, $\|W_{\nu-1}W_{\nu-1}^+p_k\| \le \|W_{k-1}W_{k-1}^+p_k\|$ (with $W_0W_0^+ \equiv 0$). But $W_{k-1}W_{k-1}^+p_k = p_k - P_{k-1}p_k$, so (3c(i)) implies

(34) $$\|W_{\nu-1}W^{+}_{\nu-1}p_k\| \le \xi\delta\|p_k\|.$$

Because $w_\nu^T P_{k-1} = (P_{k-1}w_\nu)^T = 0$, (3c(i)) also implies

(35) $$\|w_\nu^T p_k\| \le \xi\delta\|p_k\|\|w_\nu\|.$$

Now $$f'_\nu(y_\nu)p_k = (f'_\nu(y_\nu)-f'_\nu(y_{\nu-1}))P_{\nu-1}p_k + f'_\nu(y_\nu)W_{\nu-1}W^{+}_{\nu-1}p_k + w_\nu^T p_k + \\ + (f'_\nu(y_{\nu-1})P_{\nu-1}-w_\nu^T)p_k,$$

so, since $f'_\nu(y_{\nu-1})P_{\nu-1} - w_\nu^T = [P_{\nu-1}\nabla f_\nu(y_{\nu-1})-w_\nu]^T$ and

$\|(f'_\nu(x)-f'_\nu(y))P_{\nu-1}\| \le \|f'_\nu(x)-f'_\nu(y)\| \le \|\underline{f}'(x)-\underline{f}'(y)\|$, from (32), (34), (35), and (3f) we obtain

(36) $$|f'_\nu(y_\nu)p_k| \le \|p_k\|\left[\kappa\|y_\nu-y_{\nu-1}\|^\mu + \xi(\delta\|f'_\nu(y_\nu)\| + \delta\|w_\nu\|+\epsilon_3)\right].$$

Since (3f) and (27) imply

$$\|w_\nu\| \le \|P_{\nu-1}\nabla f_\nu(y_{\nu-1})\| + \epsilon_3\xi \le K_1 + \epsilon_3\xi_0,$$

if

(37) $$K_5 = 2\delta K_1 + \epsilon_3(1+\delta\xi_0),$$

then from (36) and (27) we have

(38) $$|f'_\nu(y_\nu)p_k| \le \|p_k\|[\kappa\|y_\nu-y_{\nu-1}\|^\mu + K_5\xi].$$

Now $$f'_\nu(y_\nu)(y_k-y_{k-1}) = f'_\nu(y_\nu)\left[y_k-y_{k-1}+\frac{f_k(y_{k-1})}{\pi_k}p_k\right] - \left(\frac{f'_\nu(y_\nu)p_k}{\pi_k}\right)f_k(y_{k-1}),$$

so (3e), (27), and (38) imply

$$(39)\qquad |f'_\nu(y_\nu)(y_k-y_{k-1})| \le K_1\xi\epsilon_2\|f(y_0)\| + \\ + \left\|\frac{p_k}{\pi_k}\right\| \left[\kappa\|y_\nu-y_{\nu-1}\|^\mu + K_5\xi\right] |f_k(y_{k-1})|.$$

Since (10) and (28) imply $\|\frac{p_k}{\pi_k}\| \le K_2$, where K_2 is given in (29), (33) follows from (39). ∎

We may now collect the various estimates given to show that if (32) holds and $y_0,\ldots,y_n \in B(x^*,\rho_1)$ when Algorithm (3) runs, then $\|\underline{f}(y_n)\| = \mathcal{O}(\|\underline{f}(y_0)\|\xi + \kappa\|\underline{f}(y_0)\|^{1+\mu})$ and similarly $\|y_n-x^*\| = \mathcal{O}(\|y_0-x^*\|\xi+\kappa\|y_0-x^*\|^{1+\mu})$.

Thus if we arrange that $\xi = \mathcal{O}(\|\underline{f}(y_0)\|^\mu)$, then we obtain a zero finding algorithm with local convergence of Q-order at least $1+\mu$. If $\mu=0$, then the continuity of $\underline{f}'$ implies that κ may be arbitrarily small for suitably small ρ_1, so if $\xi\to 0$ as $\|\underline{f}(y_0)\|\to 0$ then we still obtain a Q-superlinearly convergent algorithm; this is the reason for explicitly displaying the dependence of various estimates on κ.

(40) Lemma: Let $\underline{f} \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ be such that $\underline{f}(x^*) = 0$ and $\underline{f}'(x^*)$ is nonsingular. Let ξ_0 and ρ_1 be as in Lemma (20) and assume for some $\mu \in [0,1]$ and $\kappa \ge 0$ that (32) holds, i.e.

$$(41)\qquad \|\underline{f}'(x)-\underline{f}'(y)\| \le \kappa\|x-y\|^\mu \quad \forall\ x,y \in B(x^*,\rho_1).$$

There exist constants K_6, K_7, K_8, K_9 such that if Algorithm (3) runs with $y_0,\ldots,y_n \in B(x^*,\rho_1)$ and $\xi \le \xi_0$, then

(42) $$\|\underline{f}(y_n)\| \le (K_6\xi + K_7\kappa\|\underline{f}(y_0)\|^{\mu})\|f(y_0)\|$$

and

(43) $$\|y_n-x^*\| \le (K_8\xi + K_9\kappa\|y_0-x^*\|^{\mu})\|y_0-x^*\|.$$

Proof: We shall make some crude but adequate estimates in terms of $K_0,\ldots,K_4$ from Lemma (20) and K_5 from Lemma (31). Setting $Q=[q_1,\ldots,q_n]$ we have $f \equiv Q^T\underline{f} = (f_1,\ldots,f_n)^T$ and $\|f(x)\| = \|\underline{f}(x)\|\ \forall\ x$, so we may simply work with f. We now aim to show that an estimate similar to (42) holds for each component f_ν of f. To accomplish this, we note from the MVT that there exists $\theta \in (0,1)$ such that

$$f_\nu(y_n) = f_\nu(y_\nu) + f_\nu'(y_\nu+\theta[y_n-y_\nu])(y_n-y_\nu);$$

using (41) we thus obtain the estimate

(44) $$|f_\nu(y_n)| \le |f_\nu(y_\nu)| + |f_\nu'(y_\nu)(y_n-y_\nu)| + \kappa\|y_n-y_\nu\|^{1+\mu}.$$

We next obtain estimates for each term in RHS(44).

From (9), (28), and (29) we have

(45) $$|f_\nu(y_\nu)| \le K_1\epsilon_2\xi\|f(y_0)\| + K_2|f_\nu(y_{\nu-1})|(\kappa\|y_\nu-y_{\nu-1}\|^{\mu} + \epsilon_1\xi).$$

From (22) and (24) we see that

(46a) $$|f_k(y_{k-1})| \le K_2^{-1}K_{10}\|f(y_0)\|, \qquad 1 \le k \le n, \quad \text{where}$$

(46b) $$K_{10} = K_0K_1K_2 + K_3^{n-1}K_4.$$

Thus if

$$K_{14} = K_1\epsilon_2 + \epsilon_1 K_{10}$$

and

$$K_{15} = K_{10}(K_3^{n-1}K_4)^{\mu},$$

then we see from (45) and (23) that

$$|f_\nu(y_\nu)| \le (K_{14}\xi + K_{15}\kappa\|f(y_0)\|^{\mu})\|f(y_0)\|. \tag{47}$$

Using (33), (46), and (23), we find that if

$$K_{11} = (n-1)[K_1\epsilon_2 + K_5K_{10}]$$

and

$$K_{12} = (n-1)K_{10}(K_3^{n-1}K_4)^{\mu},$$

then

$$\begin{aligned} |f'_\nu(y_\nu)(y_n - y_\nu)| &\le \sum_{k=\nu+1}^{n} |f'_\nu(y_\nu)(y_k - y_{k-1})| \\ &\le (K_{11}\xi + K_{12}\kappa\|f(y_0)\|^{\mu})\|f(y_0)\|. \end{aligned} \tag{48}$$

Finally, from (23) and (25), if

$$K_{13} = \left(\frac{K_4K_3^{n-1}}{K_1K_2}\right)^{1+\mu},$$

then

$$\|y_n - y_\nu\|^{1+\mu} \le K_{13}\|f(y_0)\|^{1+\mu}, \quad 1 \le \nu \le n. \tag{49}$$

Plugging (47), (48), and (49) into (44), we obtain

$$|f_\nu(y_n)| \le [(K_{14}+K_{11})\xi + (K_{15}+K_{12}+K_{13})\kappa\|f(y_0)\|^{\mu}]\|f(y_0)\|. \tag{50}$$

Thus $\max\{|f_\nu(y_n)|: 1 \le \nu \le n\} \le \mathrm{RHS}(50)$; since

$$\|\underline{f}(y_n)\| = \|f(y_n)\| \le \sqrt{n}\max\{|f_\nu(y_n)|: 1 \le \nu \le n\},$$

(42) holds with $K_6 = (K_{14}+K_{11})\sqrt{n}$ and $K_7 = (K_{15}+K_{12}+K_{13})\sqrt{n}$. By (42), (21), and (22), (43) holds with $K_8 = K_0K_1K_6$ and $K_9 = K_0K_1^{1+\mu}K_7$. ∎

Except for Craig's method, which, under exact arithmetic, runs Algorithm (3) with $\xi = 0$, the generalizations of Brown's method considered here use finite differences to determine p_k, π_k, w_k. For example, in Brown's method $\pi_k = \frac{\Delta f_k}{\Delta p_k}(y_{k-1},h)$ (see (1.2.1)). All of the generalizations save Craig's method determine a basis $\{u_k^{(k)},\ldots,u_n^{(k)}\}$ for $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$, compute finite difference approximations $\omega_j^{(k)}$ to $\frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1}) = f_k'(y_{k-1})u_j^{(k)}$ and implicitly choose

(51) $$w_k = U_k(U_k^TU_k)^{-1}\begin{pmatrix} w_k^{(k)} \\ \vdots \\ w_n^{(k)} \end{pmatrix},$$

where

(52) $$U_k = [u_k^{(k)},\ldots,u_n^{(k)}].$$

Since

(53) $$\mathrm{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\} = \mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp},$$

U_k has full column rank, whence $U_k^TU_k$ is nonsingular, and if the $\omega_j^{(k)}$ were exact, i.e., if $\omega_j^{(k)} = \frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})$, $k \le j \le n$, then the w_k given by (51) would equal $P_{k-1}\nabla f_k(y_{k-1})$, as desired.

Algorithm (54), the generalized version of Brown's method stated below, makes use of a basis $\{u_k^{(k)},\ldots,u_n^{(k)}\}$ for $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$ to determine w_k. To prove fast local convergence, we must require that $U_k = [u_k^{(k)},\ldots,u_n^{(k)}]$ not be too poorly conditioned: thus step (54i) requires that the least singular value of U_k be not less than an arbitrary but fixed constant c_0^{-1}. We must also insist that only small errors be made in the calculations, at least near a nonsingular zero x^* of $\underline{f}$. For convenience later, we require that an error tolerance $\xi = \Xi(x_{i-1})$ be computed (implicitly) at the beginning of each major iteration, and that the approximation errors incurred during the major iteration be bounded by multiples of ξ. Often we will only be able to guarantee that particular versions of Algorithm (54) behave well in a certain neighborhood of x^*, so we allow Ξ to assume the value $+\infty$ outside this neighborhood.

If $\Xi(x) = \mathcal{O}(\|\underline{f}(x)\|^{\mu})$ near x^* for some $\mu \in [0,1]$, and if $\underline{f}'$ satisfies a Lipschitz condition of order μ near x^*, i.e. there exist $\kappa \in [0,\infty)$ and $\rho \in (0,\infty]$ such that $\|\underline{f}'(x) - \underline{f}'(y)\| \le \kappa\|x-y\|^{\mu}$ for all $x,y \in B(x^*,\rho)$, then we will be able to prove local convergence to x^* of Q-order $1+\mu$.

It is intended that finite differences be used to calculate the approximations to derivatives called for in Algorithm (54). To assure $\Xi(x) = \mathcal{O}(\|f(x)\|^{\mu})$ in this case, it suffices to arrange that the differencing step size h used satisfy $h = \mathcal{O}(\|\underline{f}(x_{i-1})\|)$ during the i-th major iteration--and that the arithmetic be sufficiently accurate (see §2.3). Except where otherwise stated, we assume exact arithmetic.

(54) <u>Algorithm</u>: [Generalized Brown's method];

(54a) Let $\underline{f} \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $\Xi: \mathbb{R}^n \to [0,\infty]$, and the constants $c_0, \delta, \epsilon_1, \epsilon_2, \epsilon_4, \eta$ be given;

OUTERLOOP:

DO $i = 1$ TO ∞;

(54b) If $\underline{f}(x_{i-1}) = 0$ THEN Stop and let $x_j = x_{i-1}$ for $j \geq i$;

(54c) $y_0 \leftarrow x_{i-1}$;

(54d) $\xi \leftarrow \Xi(y_0)$;

INNERLOOP:

DO $k = 1$ TO n;

(54e) Choose $q_k \in \mathrm{Span}\{q_1,\ldots,q_{k-1}\}^{\perp}$ with $\|q_k\| = 1$ and let $f_k \equiv q_k^T\underline{f}$;
Let $W_{k-1} = [w_1,\ldots,w_{k-1}]$ and $P_{k-1} = I - W_{k-1}W_{k-1}^{+}$, as in (1);

(54f) Choose $p_k \neq 0$ with

(54f(i)) $$\|p_k - P_{k-1}p_k\| \leq \delta\xi\|p_k\| \quad \text{and}$$

(54f(ii)) $$\left|\frac{\partial f_k}{\partial p_k}(y_{k-1})\right| \geq \eta\|P_{k-1}\nabla f_k(y_{k-1})\|\|p_k\|;$$

(54g) Find π_k with $\left|\pi_k - \frac{\partial f_k}{\partial p_k}(y_{k-1})\right| \leq \epsilon_1\xi\|p_k\|$;

(54h) Choose $y_k \in \bar{B}\left(y_{k-1} - \left[\frac{f_k(y_{k-1})}{\pi_k}\right]p_k,\ \epsilon_2\xi\|\underline{f}(y_0)\|\right)$;

(54i) Pick a basis $\{u_k^{(k)},\ldots,u_n^{(k)}\}$ for $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$ such that

$$\chi(U_k) \le c_0\|U_k\|, \qquad \text{where}$$

$$U_k = [u_k^{(k)},\ldots,u_n^{(k)}], \quad \text{as in (52);}$$

(54j) For $k \le j \le n$, find $\omega_j^{(k)}$ with

$$\left|\omega_j^{(k)} - \frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})\right| \le \epsilon_4\xi;$$

(54k) $$w_k \leftarrow U_k(U_k^T U_k)^{-1}\begin{pmatrix}\omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)}\end{pmatrix};$$

END INNERLOOP

(54ℓ) $x_i \leftarrow y_n$;

END OUTERLOOP and Algorithm (54);

Most of the versions of Algorithm (54) discussed here do not actually perform step (54k). Instead they use $\omega_k^{(k)},\ldots,\omega_n^{(k)}$ to determine $u_{k+1}^{(k+1)},\ldots,u_n^{(k+1)}$. Moreover, they choose p_k as a linear combination of $u_k^{(k)},\ldots,u_n^{(k)}$; note that (54i,j,k) do not depend on (54f,g,h), so that the former could be done after (54e). The versions of (54) proposed in §3 have the option of computing $\omega_j^{(k)}$ at y_k rather than y_{k-1}, but (54j) will still hold when this option is exercised. If (41) holds, $y_0,\ldots,y_n \in B(x^*,\rho_1)$, and h is sufficiently small, then Lemma (1.2.11) shows that the first-order finite difference approximations $\frac{\Delta f_k}{\Delta u}$ and $\frac{\check{\Delta} f_k}{\Delta u}$ defined in (1.2.1,2) may be used to compute π_k and $\omega_j^{(k)}$ in such a way that (54g) and (54j) hold, provided $\|p_k\|$ and $\|u_j^{(k)}\|$ are bounded.

Aside from the determination of w_k, the inner loop of Algorithm (54) reduces to Algorithm (3). The next lemma exhibits an ϵ_3 such that the w_k given by (54k) satisfies (3f).

(55) Lemma: In Algorithm (54), the w_k produced by step (54k) satisfies

(56) $$\|w_k - P_{k-1}\nabla f_k(y_{k-1})\| \le c_0 \sqrt{n-k+1}\, \epsilon_4 \xi,$$

so that if

(57) $$\epsilon_3 = c_0 \epsilon_4 \sqrt{n},$$

then

(58) $$\|w_k - P_{k-1}\nabla f_k(y_{k-1})\| \le \epsilon_3 \xi, \qquad 1 \le k \le n.$$

Proof: Fix i,k and let $v_j = \dfrac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})$, $k \le j \le n$. Because of (53),

$$P_{k-1} = U_k U_k^+ = U_k(U_k^T U_k)^{-1} U_k^T; \quad \text{since} \quad U_k^T \nabla f_k(y_{k-1}) = \begin{pmatrix} v_k \\ \vdots \\ v_n \end{pmatrix},$$

we thus have $P_{k-1}\nabla f_k(y_{k-1}) = U_k(U_k^T U_k)^{-1}\begin{pmatrix} v_k \\ \vdots \\ v_n \end{pmatrix}$, whence

(59) $$w_k - P_{k-1}\nabla f_k(y_{k-1}) = U_k(U_k^T U_k)^{-1}\begin{pmatrix} \omega_k^{(k)} - v_k \\ \vdots \\ \omega_n^{(k)} - v_n \end{pmatrix}.$$

By (54i), the smallest singular value σ_{min} of U_k satisfies $\sigma_{min} = \|U_k\|/\chi(U_k) \ge c_0^{-1}$. From the singular values decomposition theorem we see that

$\|U_k(U_k^TU_k)^{-1}\| = 1/\sigma_{min}$, whence

(60) $$\|U_k(U_k^TU_k)^{-1}\| \le c_0 .$$

Because $v_j = \dfrac{\partial f_k(y_{k-1})}{\partial u_j^{(k)}}$, (54j) implies

(61) $$\left\| \begin{pmatrix} \omega_k^{(k)} - v_k \\ \vdots \\ \omega_n^{(k)} - v_n \end{pmatrix} \right\| \le \epsilon_4 \xi \sqrt{n-k+1} .$$

Thus (56) follows from (59), (60), and (61). Finally, (58) follows immediately from (56) and (57). ∎

We can now prove the main result of this subsection.

(62) Theorem: Suppose $\underline{f} \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ satisfy $\underline{f}(x^*) = 0$ with $\underline{f}'(x^*)$ nonsingular. Let the constants $c_0 \in (0,\infty)$, $\eta \in (0,1]$, $\delta, \epsilon_1, \epsilon_2, \epsilon_4 \in [0,\infty)$ be fixed and let $\Xi: \mathbb{R}^n \to [0,\infty]$ be given. There exist $\rho_0, \xi_1 \in (0,\infty)$, both independent of Ξ, such that if

(63) $$\sup \Xi(B(x^*,\rho)) \le \xi_1 \quad \text{for some } \rho \in (0,\rho_0],$$

then when Algorithm (54) runs with $x_0 \in B(x^*,\rho)$, the sequence $x_0, x_1, \ldots$ of major iterates produced converges at least linearly to x^*. Moreover if Ξ vanishes and is continuous at x^*, i.e. if

(64) $$\Xi(x^*) = \lim_{\|x-x^*\|\to 0} \Xi(x) = 0,$$

and if $\mu \in [0,1]$ is such that either $\mu = 0$ or there exist $\rho_2 \in (0,\infty]$, $\kappa, \kappa_0 \in [0,\infty)$ such that

(65a) $$\|\underline{f}'(x)-\underline{f}'(y)\| \leq \kappa\|x-y\|^{\mu} \qquad \forall\ x \in B(x^*,\rho_2)$$

(65b) $$\Xi(x) \leq \kappa_0\|f(x)\|^{\mu} \qquad \forall\ x \in B(x^*,\rho_2)$$

then there exists $\rho > 0$ such that if Algorithm (54) runs with $x_0 \in B(x^*,\rho)$, then the sequence of major iterates $x_0, x_1, \ldots$ produced converges to x^* with Q-order $O_Q(x_0,x_1,\ldots) \geq 1 + \mu$. If $\mu = 0$, then the convergence is superlinear, while if $\mu > 0$, then it is strong in the sense that

(66) $$\sup\left\{\frac{\|x_i-x^*\|}{\|x_{i-1}-x^*\|^{1+\mu}}\ \middle|\ x_{i-1} \neq x^*,\ i \geq 0\right\} < +\infty.$$

Proof: The continuity of $\underline{f}'$ implies that for any $\kappa > 0$ there exists $\rho_2 > 0$ such that (65a) holds with $\mu = 0$.

With $\epsilon_3 = c_0\epsilon_4\sqrt{n}$, as in (57), let ξ_0, ρ_1, and $K_0,\ldots,K_4$ be given by Lemma (20). Recall that $\xi \leq \Xi(x_{i-1})$ during the i-th major iteration. By Lemmas (55) and (40) there exist constants K_8 and K_9 such that

(67) $$\|x_i-x^*\| \leq (K_8\Xi(x_{i-1}) + K_9\kappa\|x_{i-1}-x^*\|^{\mu})\|x_{i-1}-x^*\|,$$

whenever

(68) $$\Xi(x_{i-1}) \leq \xi_0,$$

$\mu \in [0,1]$, $\kappa \in [0,\infty)$, and $\rho_2 \in (0,\infty]$ are such that (65a) holds, and the minor iterates $y_0 = x_{i-1}, y_1,\ldots,y_n = x_i$ generated from x_{i-1} satisfy

(69) $$y_0,\ldots,y_n \in B(x^*,\min\{\rho_1,\rho_2\}).$$

Let

(70) $$\rho_3 = \min\{\rho_1,\rho_2\}[K_0K_1 + K_3^nK_4/K_2]^{-1};$$

then by (24) and (22), (69) holds whenever (65a) and (68) are valid and $x_{i-1} \in B(x^*,\rho_3)$. Thus (67) holds and $x_i \in B(x^*,\min\{\rho_1,\rho_2\})$ whenever (65a) and (68) are valid and $x_{i-1} \in B(x^*,\rho_3)$.

Let $\xi_1,\kappa > 0$ be such that $K_8\xi_1 + K_9\kappa \le 1/2$. By continuity of $\underline{f}'$ there exists $\rho_2 > 0$ such that (65a) holds with $\mu = 0$. Thus if $\rho_0 =$ RHS(70) and $\rho \in (0,\rho_0]$ is such that (63) holds, then by induction on (67) with $\mu = 0$ we have $\|x_i-x^*\| \le 2^{-i}\|x_0-x^*\|$ whenever $x_0 \in B(x^*,\rho)$, and the major iterates converge at least linearly to x^*.

Assume now that (64) holds, and let ρ_0,ξ_1 be as above. Then (64) implies the existence of $\rho \in (0,\rho_0]$ such that (63) holds, so $x_0,x_1,\ldots$ converge to x^* whenever $x_0 \in B(x^*,\rho)$.

Let $x_0,x_1,\ldots$ be a sequence generated by Algorithm (54) and converging to x^*: still assuming that (64) holds, we now prove the claims about the rate of convergence.

First suppose $\mu = 0$. For each $c>0$, (64) and the continuity of f' imply the existence of $\rho_2=\rho_2^{(c)}$ such that (65a) holds with $\kappa = 1/(2cK_9)$ and $\Xi(x_{i-1}) \le \min\{\xi_0, 1/(2cK_8)\}$ whenever $x_{i-1} \in B(x^*,\rho_2^{(c)})$. Let $\rho_3^{(c)} =$ RHS(70) with $\rho_2 = \rho_2^{(c)}$. Since $x_0,x_1,\ldots$ converge to x^*, there exists i_c such that $x_{i_c} \in B(x^*,\rho_3^{(c)})$. For $c \in (0,1)$, induction on (67) shows that $\|x_{i+1}-x^*\| \le c\|x_i-x^*\|$ for all $i \ge i_c$. Thus $x_0,x_1,\ldots$ converge superlinearly to x^*.

Finally, assume (65) holds with $\mu \in (0,1]$. Lemma (20) and (65a) imply $\Xi(x) \le \kappa_0K_0^{\mu}\|x-x^*\|^{\mu}$ whenever $x \in B(x^*,\min\{\rho_1,\rho_2\})$. Thus if ρ_3

is given by (70), then we see from (67) that

$$\|x_i - x^*\| \leq (K_8 \kappa_0 K_0^{\mu} + K_9 \kappa)\|x_{i-1} - x^*\|^{1+\mu}$$

whenever $x_{i-1} \in B(x^*, \rho_3)$, whence $\sigma_Q(x_0, x_1, \ldots) \geq 1 + \mu$ and (66) holds. ∎

In later subsections we shall use Theorem (62) to show that various versions of Algorithm (54) boast fast local convergence.

We close this subsection with a pathological note: the continuity of $\underline{f}'$ does not imply the existence of μ, κ, $\rho_2 > 0$ such that (65a) holds. In fact, we now construct an $\underline{f} \in C^1(\mathbb{R}^1)$ for which no such μ, ρ_2, κ exist. Let ϕ: $[-1/2, 1/2] \to \mathbb{R}$ be defined by

$$\phi(x) = \begin{cases} |x|^{1/\sqrt{-\ln|x|}} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}.$$

Since $\ln \phi(x) = -\sqrt{-\ln|x|}$, $0 < |x| \leq 1/2$, $\lim_{x \to 0} \phi(x) = 0$ and ϕ is continuous. However, there exist no μ, $\rho_2, \kappa > 0$ such that $|\phi(x) - \phi(0)| = \phi(x) \leq \kappa|x|^{\mu}$ $\forall$ $x \in B(0, \rho_2)$. Indeed, for any $\mu > 0$ and x with $0 < |x| \leq 1/2$ we have

$$\ln\left[\frac{\phi(x)}{|x|^{\mu}}\right] = -\sqrt{-\ln|x|} - \mu \ln|x|,$$

whence

$$\lim_{x \to 0} \frac{\phi(x)}{|x|^{\mu}} = +\infty.$$

Now let $\underline{f}(x) = x + \int_0^x \phi(y)dy$ for $|x| \leq 1/2$, and let $\underline{f}$ be defined on $\mathbb{R} - [-1/2, 1/2]$ in any manner such that $\underline{f} \in C^1(\mathbb{R}^1)$. For $|x| < 1/2$ we have $\underline{f}'(x) = 1 + \emptyset(x)$, whence $\underline{f}'(x) - \underline{f}'(0) = \emptyset(x)$, so for $x^* = 0$ there do not exist $\mu > 0$, $\rho_2, \kappa > 0$ such that (65a) holds.

I suspect that if Algorithm (54) runs with the above $\underline{f}$, with π_k approximated by forward differences and $x_0 > 0$ sufficiently small, then the sequence $x_0, x_1, \ldots$ of major iterates here has $\sigma_Q(x_0, x_1, \ldots) = 1$, despite the fact that this sequence converges Q-superlinearly to 0 by Theorem (62).

Using the above example, it is easy to construct $\underline{f} \in C^1(\mathbb{R}^n)$ such that (65a) does not hold for any $\mu > 0$.

Of course, for most functions $\underline{f}$ of practical interest there exist $\kappa, \rho_2 > 0$ such that (65a) holds with $\mu = 1$.

2.2 Brown's Method

In this subsection we examine Brown's method (Brown, 1966) in some detail and show that it is a particular version of Algorithm (2.1.54). We also see how to implement it so that the housekeeping takes only $n^2/4 + O(n)$ storage locations and $O(n^3)$ arithmetic operations per major iteration, rather than the $n^2/2 + O(n)$ storage locations and $O(n^4)$ arithmetic operations required when the method is implemented as Brown has suggested (1966), (1973). Since this method always chooses $q_k = e_k$ in step (2.1.54e), whence $Q = [q_1, \ldots, q_n] \equiv I$ and $f \equiv Q^T\underline{f} = \underline{f}$, we will refer to f rather than $\underline{f}$ in this subsection.

Brown derived his algorithm by a different line of reasoning than that presented above. The view presented here was suggested in a paper by Brent (1973), who noted that Brown's method proceeds by constructing the linear approximations F_k to f_k discussed in §2.1 (see (2.1.2)) and selecting points y_k by projecting y_{k-1} in a certain direction onto the n-k dimensional hyperplane H_k in which $F_1,\ldots,F_k$ vanish. We now describe Brown's viewpoint, then relate his method to (2.1.54).

Brown's idea was to compute linearizations F_k of the component functions f_k of f one at a time and to use the requirement that $F_1,\ldots,$ F_k vanish to eliminate k variables. In more detail, let the major iterate x_i have components $x_i^j = e_j^T x_i$, so that $x_i = (x_i^1,\ldots,x_i^n)^T$, and regard $x = (x^1,\ldots,x^n)^T$ as a vector of variables. During the k-th minor iteration, Brown expresses $x^1,\ldots,x^k$ as linear functions $x^j = X_j^{(k)}(x^{k+1},\ldots,x^n)$ of the remaining variables. The $X_j^{(k)}$ are chosen so that for any $(x^{k+1},\ldots,x^n) \in \mathbb{R}^{n-k}$, $F_j(X_1^{(k)}(x^{k+1},\ldots,x^n),\ldots,X_k^{(k)}(x^{k+1},\ldots,x^n),x^{k+1},\ldots,x^n) = 0$, $1 \le j \le k$. Before constructing $X_1^{(k)},\ldots,X_k^{(k)}$, Brown reorders the variables ("pivots") to guarantee stability. From his viewpoint, his algorithm may be stated in the following form (in which, for notational simplicity, major iterate x_{i+1} rather than x_i is determined during the i-th major iteration).

(1) Algorithm: [Brown's method from his viewpoint];

(1a) Let $f \in C^1(\mathbb{R}^n)$, $x_1 \in \mathbb{R}^n$, $\gamma > 0$ be given;

MAJORLOOP:

DO i = 1 to ∞;

(1b) If $f(x_i) = 0$ then stop and let $x_j = x_i$ for $j > i$;

(1c) Choose $h \in (0,\gamma\|f(x_i)\|]$;

MINORLOOP:

DO k = 1 TO n;

(1d) Let $\bar{f}_k: \mathbb{R}^{n-k+1} \to \mathbb{R}$ be defined by

$$\bar{f}_k(x^k,\ldots,x^n) = f_k(X_1^{(k-1)}(x^k,\ldots,x^n),\ldots,X_{k-1}^{(k-1)}(x^k,\ldots,x^n), x^k,\ldots,x^n);$$

[Note that $\bar{f}_1 = f_1$];

(1e) For $k \leq j \leq n$, let

$$\omega_j^{(k)} = \frac{\bar{f}_k(x_i^k,\ldots,x_i^{j-1},x_i^j+h,x_i^{j+1},\ldots,x_i^n)-\bar{f}_k(x_i^k,\ldots,x_i^n)}{h};$$

(1f) Reorder the variables if necessary to arrange that

$$|\omega_k^{(k)}| = \max\{|\omega_j^{(k)}|: \ k \leq j \leq n\};$$

(1g) Let $X_k^{(k)}(x^{k+1},\ldots,x^n) = x_i^k - \left[\frac{\bar{f}_k(x_i^k,\ldots,x_i^n)}{\omega_k^{(k)}} + \sum_{j=k+1}^{n}\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)(x^j-x_i^j)\right];$

(1h) For $1 \leq j \leq k-1$, let

$$X_j^{(k)}(x^{k+1},\ldots,x^n) = X_j^{(k-1)}(X_k^{(k)}(x^{k+1},\ldots,x^n),x^{k+1},\ldots,x^n);$$

[Note that $X_1^{(n)},\ldots,X_n^{(n)}$ are constants];

END INNERLOOP;

(1i) $x_{i+1} \leftarrow (X_1^{(n)},\ldots,X_n^{(n)})^T;$

END OUTERLOOP and Algorithm (1);

Notice in the inner loop of (1) that

$$\bar{F}_k(x^k,\ldots,x^n) \equiv \bar{f}_k(x_i^k,\ldots,x_i^n) + \sum_{j=k}^{n} \omega_j^{(k)}(x^j-x_i^j)$$

is an approximate first-order Taylor expansion of $\bar{f}_k$ about $(x_i^k,\ldots,x_i^n)$, since $\omega_j^{(k)}$ approximates

$$\frac{\partial \bar{f}_k}{\partial x^j}(x^k,\ldots,x^n)\Big|_{(x^k,\ldots,x^n)=(x_i^k,\ldots,x_i^n)} .$$

Brown had this in mind and chose $X_k^{(k)}$ as in (1g) so that

$$\bar{f}_k(X_k^{(k)}(x^{k+1},\ldots,x^n),x^{k+1},\ldots,x^n) = 0 \quad \forall\, (x^{k+1},\ldots,x^n) \in \mathbb{R}^{n-k}.$$

We now examine in more detail how the dependent component functions $X_j^{(k)}$ may be evaluated. This will lead to mathematically equivalent reformulations of Algorithm (1) in terms of (2.1.54).

Fix i. Let $\omega_j^{(k)}$ be produced by the inner loop of Algorithm (1) and define $u_j^{(k)}$ by:

(2a) $$u_j^{(1)} = e_j, \quad 1 \leq j \leq n \quad \text{and}$$

(2b) $$u_j^{(k+1)} = u_j^{(k)} - \left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right) u_k^{(k)}, \quad k < j \leq n,\ 1 \leq k \leq n.$$

Given these $u_j^{(k)}$, define y_k by:

(3a) $$y_0 = x_i \quad \text{and}$$

(3b) $$y_k = y_{k-1} - \left(\frac{f_k(y_{k-1})}{\omega_k^{(k)}}\right) u_k^{(k)}, \quad 1 \leq k \leq n.$$

For $1 \leq k \leq n+1$, let $X^{(k-1)}: \mathbb{R}^{n-k+1} \to \mathbb{R}^n$ be given by

(4) $$X^{(k-1)}(x^k,\ldots,x^n) = (X_1^{(k-1)}(x^k,\ldots,x^n),\ldots,X_{k-1}^{(k-1)}(x^k,\ldots,x^n), x^k,\ldots,x^n)^T,$$

with $X^{(0)}(x^1,\ldots,x^n) = (x^1,\ldots,x^n)^T$ and $X^{(n)} = (X_1^{(n)},\ldots,X_n^{(n)})^T$, a constant.

From (1d) we see

(5) $$\bar{f}_k = f_k \circ X^{(k-1)}, \quad \text{i.e.} \quad \bar{f}_k(x^k,\dots,x^n) = f_k(X^{(k-1)}(x^k,\dots,x^n)).$$

Now $X^{(0)}(x^1,\dots,x^n) = y_0 + \sum_{j=1}^{n} (x^j - x_i^j)u_j^{(1)}$, by (2a), (3a), and (4). Assume inductively that we have k with $1 \le k \le n$ such that

(6) $$X^{(k-1)}(x^k,\dots,x^n) = y_{k-1} + \sum_{j=k}^{n} (x^j - x_i^j)u_j^{(k)}.$$

From (1e), (5), (6), and (1.2.1) we obtain

(7) $$\omega_j^{(k)} = \frac{f_k(y_{k-1}+hu_j^{(k)}) - f_k(y_{k-1})}{h} \equiv \frac{\Delta f_k}{\Delta u_j^{(k)}}(y_{k-1};h).$$

Furthermore, (1h), (1g), (4), (5), and (6) imply

$$\begin{aligned} X^{(k)}(x^{k+1},\dots,x^n) &= y_{k-1} + (X_k^{(k)}(x^{k+1},\dots,x^n) - x_i^k)u_k^{(k)} + \\ &\quad + \sum_{j=k+1}^{n} (x^j - x_i^j)u_j^{(k)} \\ &= \left(y_{k-1} - \left[\frac{f_k(y_{k-1})}{\omega_k^{(k)}} \right] u_k^{(k)} \right) + \\ &\quad + \sum_{j=k+1}^{n} (x^j - x_i^j) \left(u_j^{(k)} - \left[\frac{\omega_j^{(k)}}{\omega_k^{(k)}} \right] u_k^{(k)} \right). \end{aligned}$$

Therefore, by (2b) and (3b), (6) holds with k replaced by $k+1$. Hence (6) holds for $1 \le k \le n+1$ and $x_{i+1} \equiv (X_1^{(n)},\dots,X_n^{(n)})^T = y_n$. We thus see that Brown's method (1) is mathematically equivalent to the version of (2.1.54) in which $u_j^{(k)}$ and $\omega_j^{(k)}$ are determined by (2) and (7) respectively, subscripts are interchanged if need be to insure $|\omega_k^{(k)}| = \max\{|\omega_j^{(k)}|: k \le j \le n\}$, $p_k = u_k^{(k)}$, and $\pi_k = \omega_k^{(k)}$.

In the reasoning just presented we suggest computing $u_j^{(k)}$ by (2), then using (6) to evaluate $X^{(k-1)}(x^k,\ldots,x^n)$. This appears to be a new way to arrange the calculations in Brown's method, for Brown had in mind using (1h) and (1g) to determine $X^{(k-1)}$ recursively via the following

(8) <u>Algorithm</u>: [to evaluate $X^{(k-1)}(x^k,\ldots,x^n)$];

(8a) Let $x_i \in \mathbb{R}^n$, $(x^k,\ldots,x^n)^T \in \mathbb{R}^{n-k+1}$, the ratios $\left(\dfrac{\omega_j^{(\ell)}}{\omega_\ell^{(\ell)}}\right)$ for $1 \le \ell < k$ and $\ell < j \le n$, and the previously computed values

$$x_i^\ell - \frac{\bar{f}_\ell(x_i^\ell,\ldots,x_i^n)}{\omega_\ell^{(\ell)}} = x_i^\ell - \frac{f_\ell(y_{\ell-1})}{\omega_\ell^{(\ell)}}, \quad 1 \le \ell < k, \text{ be given;}$$

DO ℓ = k-1 TO 1 BY -1;

(8b)
$$x^\ell \leftarrow \left(x_i^\ell - \frac{f_\ell(y_{\ell-1})}{\omega_\ell^{(\ell)}}\right) - \sum_{j=\ell+1}^{n}\left(\frac{\omega_j^{(\ell)}}{\omega_\ell^{(\ell)}}\right)(x^j - x_i^j);$$

END;

(8c) Return $X^{(k-1)}(x^k,\ldots,x^n) = (x^1,\ldots,x^n)^T$;

END Algorithm (8);

To see the connection between (8) and (6), let

$$
(9) \quad \Omega_k = \begin{bmatrix}
1 & \frac{\omega_2^{(1)}}{\omega_1^{(1)}} & \frac{\omega_3^{(1)}}{\omega_1^{(1)}} & \cdots & \cdots & \cdots & \cdots & \frac{\omega_n^{(1)}}{\omega_1^{(1)}} \\
0 & 1 & \frac{\omega_3^{(2)}}{\omega_2^{(2)}} & \cdots & \cdots & \cdots & \cdots & \frac{\omega_n^{(2)}}{\omega_2^{(2)}} \\
\vdots & \ddots & \ddots & \ddots & & & & \vdots \\
\vdots & & \ddots & 1 & \frac{\omega_k^{(k-1)}}{\omega_{k-1}^{(k-1)}} & \frac{\omega_{k+1}^{(k-1)}}{\omega_{k-1}^{(k-1)}} & \cdots & \frac{\omega_n^{(k-1)}}{\omega_{k-1}^{(k-1)}} \\
\vdots & & & 0 & 1 & 0 & \cdots & 0 \\
\vdots & & & & \ddots & \ddots & \ddots & \vdots \\
\vdots & & & & & \ddots & 1 & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1
\end{bmatrix}
$$

with $\Omega_1 = I$. We see immediately that (8) solves the upper triangular linear system

$$
\Omega_k z = \left(x_i^1 - \frac{f_1(y_0)}{\omega_1^{(1)}}, \ldots, x_i^{k-1} - \frac{f_{k-1}(y_{k-2})}{\omega_{k-1}^{(k-1)}}, x^k, \ldots, x^n\right)^T + (\Omega_k - I)x_i
$$

for $z = (z^1, \ldots, z^n)^T$ by backwards substitution and returns z. Thus $X^{(k-1)}(x^k, \ldots, x^n)$ has the form

(10) $$X^{(k-1)}(x^k,\ldots,x^n) = \text{constant} + \Omega_k^{-1}\left(\sum_{j=k}^{n}\right)x^j e_j \quad .$$

Comparing (6) and (10), we see that

(11) $$u_j^{(k)} = \Omega_k^{-1} e_j .$$

Putting the foregoing observations together, we see that Brown's method is mathematically equivalent to

(12) <u>Algorithm</u>: [Brown's method in terms of (2.1.54)];

(12a) Let $f \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $\gamma > 0$ be given;

OUTERLOOP:

DO $i = 1$ TO ∞;

(12b) If $f(x_{i-1}) = 0$ then stop and let $x_j = x_{i-1}$ for $j \geq i$;

(12c) Choose $h \in (0, \gamma\|f(x_{i-1})\|]$;

(12d) $y_0 \leftarrow x_{i-1}$;

(12e) Let $u_j^{(1)} = e_j$, $1 \leq j \leq n$;

INNERLOOP:

DO $k = 1$ TO n;

(12f) For $k \leq j \leq n$, let $\omega_j^{(k)} = \dfrac{\Delta f_k}{\Delta u_j^{(k)}}(y_{k-1}, h)$;

(12g) Relabel if necessary to arrange that

$$|\omega_k^{(k)}| = \max\{|\omega_j^{(k)}| : k \leq j \leq n\};$$

(12h) $$y_k \leftarrow y_{k-1} - \left(\frac{f_k(y_{k-1})}{\omega_k^{(k)}}\right) u_k^{(k)};$$

[Thus let $p_k = u_k^{(k)}$, $\pi_k = \omega_k^{(k)}$];

For $k < j \leq n$, either

(12i) $$u_j^{(k+1)} \leftarrow u_j^{(k)} - \left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right) u_k^{(k)}$$

or

(12i)' $$u_j^{(k+1)} \leftarrow \Omega_{k+1}^{-1} e_j,$$

i.e. solve $\Omega_{k+1} u_j^{(k+1)} = e_j$, with Ω_{k+1} as in (9);

END INNERLOOP;

$x_i \leftarrow y_n$;

END OUTERLOOP and Algorithm (12);

When Algorithm (12) is implemented in finite precision arithmetic, step (12i) has the opportunity to make the accumulated roundoff error in $u_j^{(k+1)}$ as much as $\left(1 + \left|\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right|\right)$ times as large as that in $u_j^{(k)}$, i.e. as much as twice as large, since $|\omega_j^{(k)}/\omega_k^{(k)}|$ can be as large as 1. Thus (12i)' might appear to be numerically more stable than (12i). However, $\|\Omega_k^{-1}\|$ can be nearly twice as large as $\|\Omega_{k-1}^{-1}\|$, so the worst-case error bound which we derive in §2.3 for (12i)' is actually larger than the one we derive there for (12i). Some very scanty computational evidence suggests that (12i) may not fare much worse than (12i)' in practice. The computational evidence was obtained by using (12i) and (12i)' to calculate floating-point approximations $fl(\Omega_{n-1}^{-1})$ to $\Omega_{n-1}^{-1} = [u_1^{(1)}, u_2^{(2)}, \ldots, u_n^{(n)}]$ and then computing the residuals $\Omega_{n-1} fl(\Omega_{n-1}^{-1}) - I$. In one typical example with $n = 31$, (12i) did worse than (12i)' by a factor of only 10. In another run, which came near to generating the largest possible $\|\Omega_{n-1}^{-1}\|$, (12i) performed a trifle better than (12i)'.

From the standpoint of time and storage, (12i) has considerable advantages over (12i)'. Indeed, during one major iteration in which $\omega_j^{(k)} \neq 0$ for most $j > k$, $1 \leq k < n$, (12i) requires $\mathcal{O}(n^3)$ arithmetic operations while (12i)' requires $\mathcal{O}(n^4)$. Moreover, (12i)' requires $n^2/2 + \mathcal{O}(n)$ storage locations to hold the upper triangular portion of Ω_k, while (12i) requires only $n^2/4 + \mathcal{O}(n)$ locations for the nontrivial portions of $u_k^{(k)}, \ldots, u_n^{(k)}$. (Note that $u_j^{(k)}$ has the form $u_j^{(k)} = \begin{pmatrix} \bar{u}_j^{(k)} \\ \cdots\cdots \\ e_{j-k+1} \end{pmatrix}$, where $\bar{u}_j^{(k)} \in \mathbb{R}^{k-1}$). Of course, in many practical applications $n^2/4$ will be small and the work involved in (12i) or (12i)' will be overshadowed by the effort required to evaluate f_k.

It is instructive to see how Algorithm (12) behaves when $f(x) = Ax - b$ is linear with $A \in \mathcal{L}(\mathbb{R}^n)$ nonsingular. For simplicity, assume that no pivoting is necessary, i.e. that it is never necessary to interchange subscripts in step (12g). Let

(13) $$\mathcal{U}^{(k)} = [u_1^{(1)}, u_2^{(2)}, \ldots, u_k^{(k)}, u_{k+1}^{(k)}, \ldots, u_n^{(k)}]$$

and

(14) $$L^{(k)} = A\mathcal{U}^{(k)} = [\ell_1^{(1)}, \ell_2^{(2)}, \ldots, \ell_k^{(k)}, \ell_{k+1}^{(k)}, \ldots, \ell_n^{(k)}],$$

where $\ell_j^{(\nu)} = Au_j^{(\nu)}$. As noted above, we see from (12i) that $u_j^{(k)} = \begin{pmatrix} \bar{u}_j^{(k)} \\ \cdots \\ e_{j-k+1} \end{pmatrix}$; thus $\mathcal{U}^{(k)}$ is an upper triangular matrix with ones on the main diagonal and I_{n-k+1} in the lower right-hand corner. From (12f) and (14) we see that the k-th row of $L^{(k)}$ is just

$$e_k^T L^{(k)} = (\omega_1^{(k)}, \ldots, \omega_n^{(k)}), \tag{15}$$

where $\omega_j^{(k)}$ comes from (12f) for $k \le j \le n$. From (12i) and (13) we see that

$$\mathcal{U}^{(k+1)} = \mathcal{U}^{(k)} \tilde{\mathcal{U}}^{(k)}, \tag{16a}$$

where

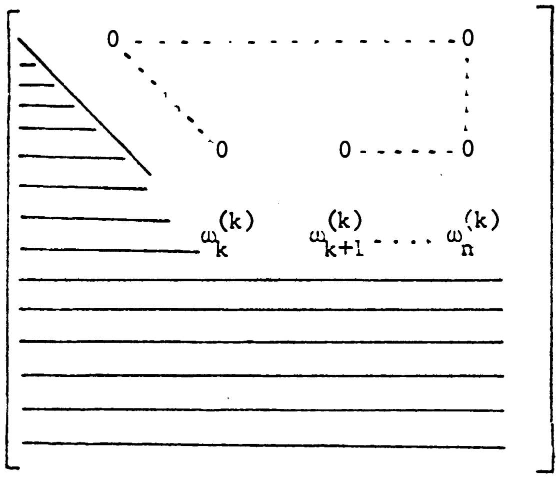
$$\tilde{\mathcal{U}}^{(k)} = \left[\begin{array}{ccccccc} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & 0 & 1 & -\dfrac{\omega_{k+1}^{(k)}}{\omega_k^{(k)}} & \cdots & -\dfrac{\omega_n^{(k)}}{\omega_k^{(k)}} \\ \vdots & & & 0 & 1 & 0 \cdots & 0 \\ \vdots & & & & \ddots & \ddots & 0 \\ 0 & \cdots & & & & 0 & 1 \end{array}\right] \begin{array}{l} \\ \\ \\ \text{k-th row} \\ \\ \\ \\ \end{array}, \tag{16b}$$

i.e.,

$$\tilde{\mathcal{U}}^{(k)} = I - e_k\left(\sum_{j=k+1}^{n}\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)e_j\right)^T.$$

Thus, by (14), (15), and (16), $L^{(k+1)} = L^{(k)}\tilde{\mathcal{U}}^{(k)}$ has zeroes in the last $n-k$ positions of row k. By induction on k we see that $e_j^T\ell_\nu^{(k)} = 0$ for $j < \min\{k,\nu\}$, whence $L^{(k)}$ has the form

$$L^{(k)} =$$



and $L^{(n)} = A\mathcal{U}^{(n)}$ is lower triangular. Since $\tilde{\mathcal{U}}^{(k)-1} = I + e_k\left(\sum_{j=k+1}^{n}\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)e_j\right)^T$ is obtained from $\tilde{\mathcal{U}}^{(k)}$ by changing $-\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)$ to $+\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)$ in (16b), we see from (9) that $\Omega_{k+1} = \tilde{\mathcal{U}}^{(k)-1}\Omega_k$. But $\Omega_1 = \mathcal{U}^{(1)} = I$, so by (16a) and induction on k, $\mathcal{U}^{(k)} = \Omega_k^{-1}$, $1 \le k \le n$. In particular, $L^{(n)} = A\Omega_n^{-1}$, so $A = L^{(n)}\Omega_n$ expresses A as the product of a lower triangular matrix and an upper triangular matrix with ones on the diagonal. Thus, when presented with the linear $f(x) = Ax-b$, Algorithm (12) performs column-wise Gaussian elimination. If relabeling is required in step (12g), then this becomes column-wise Gaussian elimination with partial pivoting. Hence Brown's method is a generalization of column-wise Gaussian elimination with partial pivoting. Note that the lower triangular portion of $L^{(n)}$ is not actually computed.

When $f(x) = Ax - b$ is linear, the linear approximations F_k to f_k are exact. Therefore $f_j(y_k) = e_j^T(Ay_k - b) = 0$ for $1 \leq j \leq k$ and $f(y_n) = 0$. If such an f (with nonsingular A) is given to Algorithm (12), then $x_i = x^*$ $\forall\ i \geq 1$.

We now state and prove several lemmas in order to show that Algorithm (12) satisfies the conditions of Algorithm (2.1.54) with $\xi = \kappa h^{\mu}$ near a nonsingular zero x^*. Thus Theorem (2.1.62) applies and insures that Brown's method has fast local convergence. To begin with, we examine condition (2.1.54f(ii)). In this regard, the following lemma will be useful here as well as later in this section.

(17) <u>Lemma</u>: Let the constants δ, ϵ_1, ϵ_4, $c_0 \in (0,\infty)$ be fixed and let $f \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ have $f(x^*) = 0$ with $f'(x^*)$ nonsingular. Assume for some $\rho_3 \in (0,\infty]$ and $\eta_1 \in (0,1]$ that $w_1,\ldots,w_{k-1}$ are generated according to steps (2.1.54i,j,k) of Algorithm (2.1.54) and that $p_k \in \mathbb{R}^n$ satisfies (2.1.54f(i)) as well as

(18) $$|p_k^T w_k| \geq \eta_1 \|p_k\| \|w_k\|$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_3)$. Then for each $\eta \in (0,\eta_1)$ there exist $\rho_4, \xi_1 > 0$ such that (2.1.54f(ii)) holds, i.e.

(19) $$\left|\frac{\partial f_k}{\partial p_k}(y_{k-1})\right| \equiv |p_k^T \nabla f_k(y_{k-1})| \geq \eta \|P_{k-1}\nabla f_k(y_{k-1})\| \|p_k\| ,$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_4)$ and $\xi \leq \xi_1$.

Proof: By Lemmas (2.1.15) and (2.1.55) and the continuity of f' there exist $\xi_0, \rho_1 \in (0,\infty)$ and $K_1 = \max\|f'(\bar{B}(x^*,\rho_1))\|$ such that

(20a) $$\|\nabla f_k(y_{k-1})\| \leq \|f'(y_{k-1})\| \leq K_1$$

and

(20b) $$\|P_{k-1}\nabla f_k(y_{k-1})\| \geq \frac{1}{2\|f'(x^*)^{-1}\|}$$

whenever

(21) $$y_0,\ldots,y_{k-1} \in B(x^*,\rho_1) \quad \text{and} \quad \xi \leq \xi_0.$$

Assume (21) holds. From (18), (2.1.54f(i)), and (2.1.58) we have

(22) $$|p_k^T\nabla f_k(y_{k-1})| \geq |p_k^T w_k| - \|p_k\|\|P_{k-1}\nabla f_k(y_{k-1})-w_k\| - \|p_k-P_{k-1}p_k\|\|\nabla f_k(y_{k-1})\|$$
$$\geq \eta\|p_k\|\|w_k\| - \epsilon_3\xi\|p_k\| - \delta\xi\|\nabla f_k(y_{k-1})\|\|p_k\|.$$

By (2.1.58) we also have

(23) $$\|w_k\| \geq \|P_{k-1}\nabla f_k(y_{k-1})\| - \|w_k-P_{k-1}\nabla f_k(y_{k-1})\|$$
$$\geq \|P_{k-1}\nabla f_k(y_{k-1})\| - \epsilon_3\xi.$$

Combining (20) and (23) with (22) and recalling that $\eta_1 \leq 1$, we obtain

(24) $$|p_k^T\nabla f_k(y_{k-1})| \geq \left(\eta_1[\|P_{k-1}\nabla f_k(y_{k-1})\| - \epsilon_3\xi] - [\epsilon_3+\delta K_1]\xi\right)\|p_k\|$$
$$\geq \left(\eta_1 - \frac{[2\epsilon_3+\delta K_1]\xi}{\|P_{k-1}\nabla f_k(y_{k-1})\|}\right)\|P_{k-1}\nabla f_k(y_{k-1})\|\|p_k\|$$

$$\geq (\eta_1 - 2[2\epsilon_3+\delta K_1]\|f'(x^*)^{-1}\|\xi)\|P_{k-1}\nabla f_k(y_{k-1})\|\|p_k\|.$$

Now let $\rho_4 = \min\{\rho_1,\rho_3\}$ and let $\eta \in (0,\eta_1)$ be given. From (24) we see that (19) is valid whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_4)$ and

$$\xi \leq \xi_1 \equiv \frac{\eta_1 - \eta}{2[2\epsilon_3 + \delta K_1]\|f'(x^*)^{-1}\|}. \quad \blacksquare$$

We will use Lemma (26), given below, to exhibit an $\eta_1 > 0$ such that (18) holds for Brown's method. The proof of Lemma (26) utilizes the following lemma with $X = \{x \in \mathbb{R}^m \mid \|x\| = 1\}$, the unit sphere of $\mathbb{R}^m$.

(25) <u>Lemma</u>: Let X be a nonempty countably compact Hausdorff space and assume that $\phi_i: X \to \mathbb{R}$ and $\psi_i: X \to \mathbb{R}$ are continuous for $1 \leq i \leq m$. If $g: X \to \mathbb{R}$ is given by

$$g(x) = \min\{\psi_i(x) \mid \phi_i(x) = \max\{\phi_j(x) \mid 1 \leq j \leq m\}\},$$

then g attains its minimum, i.e. there exists $y \in X$ such that $g(y) = \inf g(X)$.

<u>Proof</u>: Let $x_1,x_2,\ldots$ be a sequence of points in X such that

$$\lim_{i\to\infty} g(x_i) = \underline{g} \equiv \inf g(X);$$

let j_i be such that $\phi_{j_i}(x_i) = \max\{\phi_k(x_i) \; 1 \leq k \leq m\}$ and $\psi_{j_i}(x_i) = g(x_i)$ $\forall\ i \geq 1$. For some $\bar{j}$, $1 \leq \bar{j} \leq m$, we must have $\bar{j} = j_i$ for infinitely many i, so we may extract from $\{x_i \mid i \geq 1\}$ a sequence $y_1,y_2,\ldots$ with $\phi_{\bar{j}}(y_i) = \max\{\phi_j(y_i) \mid 1 \leq j \leq m\}$, $\psi_{\bar{j}}(y_i) = g(y_i)$, and $\lim_{i\to\infty} g(y_i) = \underline{g}$. Since X is countably compact, the sequence $y_1, y_2, \ldots$ has a limit point y, and WLOG $\lim_{i\to\infty} y_i = y$. By the continuity of ϕ_i and ψ_i $(1 \le i \le m)$, we must have $\phi_{\bar{j}}(y) = \max\{\phi_j(y) \mid 1 \le j \le m\}$ and $g(y) = \psi_{\bar{j}}(y) = \underline{g} \equiv \inf g(X)$. ∎

(26) Lemma: Assume that $V = [v_1, \ldots, v_m] \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$ has full column rank, so that the smallest singular value $\sigma_{\min}$ of V is positive. Let v^* be a column of V having maximum 2-norm, i.e. $\|v^*\| = \max\{\|v_j\| : 1 \le j \le m\}$. If $\underline{w} = Vy \ne 0$ lies in the column space of V and, say, $|\underline{w}^T v_1| = \max\{|\underline{w}^T v_i| : 1 \le i \le m\}$, then

$$\text{(27)} \qquad \frac{|\underline{w}^T v_1|}{\|\underline{w}\|\|v_1\|} \ge \frac{\sigma_{\min}}{\|v^*\|\sqrt{m}} \ge \frac{1}{\chi(V)\sqrt{m}} .$$

Proof: Since $\sigma_{\min} / \|v^*\| \ge \sigma_{\min} / \|V\| = 1/\chi(V)$, we need only prove the first inequality of (27).

Let Q be an $n \times n$ orthogonal matrix such that $QV = \begin{pmatrix} R \\ 0 \end{pmatrix}$, where R is an $m \times m$, necessarily nonsingular matrix, and let $w = Ry \in \mathbb{R}^m$, so that $\underline{w} = Vy = Q^T\begin{pmatrix} w \\ 0 \end{pmatrix}$. Setting $r_i = Re_i$, so that $R = [r_1, \ldots, r_m]$, we have

$$\text{(28)} \qquad \underline{w}^T v_i = w^T r_i, \quad \|\underline{w}\| = \|w\|, \text{ and } \|v_i\| = \|r_i\|, \quad 1 \le i \le m.$$

Consider the $g: (\mathbb{R}^m - \{0\}) \to \mathbb{R}$ given by

$$g(x) = \min\left\{ \frac{|x^T r_i|}{\|x\|\|r_i\|} : |x^T r_i| = \max\{|x^T r_j| : 1 \le j \le m\} \right\} .$$

If $X = \{x \in \mathbb{R}^m : \|x\| = 1\}$ and $\hat{g} = g|_X$ is the restriction of g to X, then $g(x) = \hat{g}(x/\|x\|)$ for any $x \in \mathbb{R}^m - \{0\}$. By Lemma (25) $\hat{g}$ attains its minimum, so there exists $z \in \mathbb{R}^m - \{0\}$ such that

(29) $$g(z) = \min g(\mathbb{R}^m - \{0\}).$$

Because of (28), it suffices to show that $g(z) \geq \sigma_{min} / (\|v^*\| \sqrt{m})$.

Now we must have $|z^T r_i| = |z^T r_1|$ for $1 \leq i \leq m$, since otherwise, say if

$$|z^T r_1| = \max\{|z^T r_i|: 1 \leq i \leq m\} > |z^T r_2| \quad \text{and} \quad u = R^{-T} e_2,$$

then $u^T r_i = \begin{cases} 1 & \text{if } i = 2 \\ 0 & \text{if } i \neq 2 \end{cases}$, so $|(z+\lambda u)^T r_i| = |z^T r_i|$ for $i \neq 2$ and any $\lambda \in \mathbb{R}$, while $|(z+\lambda u)^T r_2| \leq |z^T r_1|$ for $|\lambda|$ sufficiently small: but if $\lambda u^T z \geq 0$ with $|\lambda| > 0$, then $\|z+\lambda u\| > \|z\|$, so for all sufficiently small $|\lambda| > 0$ with $\lambda u^T z \geq 0$, $g(z+\lambda u) < g(z)$, which contradicts (29). Hence $|z^T R| = (|z^T r_1|, \ldots, |z^T r_m|) = \|z^T R\|_\infty e^T$, where $e = (1,1,\ldots,1)^T \in \mathbb{R}^m$. WLOG $\|z^T R\|_\infty = 1$. Then $\|z^T R\| = \sqrt{m}$, so $\|z\| = \|z^T R R^{-1}\| \leq \|R^{-1}\| \sqrt{m}$. Furthermore, $\|v^*\| = \max\{\|r_i\|: 1 \leq i \leq m\}$ and $\|R^{-1}\| = 1/\sigma_{min}$, so

$$g(z) \geq \frac{1}{\|v^*\| \|R^{-1}\| \sqrt{m}} = \frac{\sigma_{min}}{\|v^*\| \sqrt{m}}. \quad ∎$$

We now exhibit an $\eta_1 \in (0,1)$ such that (18) holds for Brown's method (with exact arithmetic).

(30) <u>Lemma</u>: If Algorithm (12) runs, then (18) holds with

(31) $$\eta_1 = \sqrt{\frac{3}{2(2+4^{n-2})}}.$$

Proof: The case $n = 1$ being trivial, assume $n > 1$. Let k satisfy $1 \le k \le n$ and assume WLOG that no relabeling is performed in step (12g). From (12e), (12i), and the fact that $|\omega_j^{(\ell)}/\omega_\ell^{(\ell)}| \le 1$ for $1 \le \ell < k$ and $\ell < j \le n$, we see by induction on k that

$$\text{(32a)} \qquad u_j^{(k)} = \begin{pmatrix} \bar{u}_j^{(k)} \\ \cdots \\ e_{j-k+1} \end{pmatrix}, \text{ where } \bar{u}_j^{(k)} \in \mathbb{R}^{k-1}$$

with

$$\text{(32b)} \qquad |e_\nu^T \bar{u}_j^{(k)}| \le 2^{k-1-\nu} \quad \text{for } 1 \le \nu \le k-1,$$

whence

$$\text{(32c)} \qquad \|u_j^{(k)}\| \le \sqrt{1 + (4^{k-1}-1)/3} = \sqrt{(2+4^{k-1})/3}, \quad k \le j \le n.$$

Let $m = n-k+1$ and $V = U_k \equiv [u_k^{(k)}, \ldots, u_n^{(k)}] \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$. Since V has the form $V = \begin{pmatrix} \bar{V} \\ I_m \end{pmatrix}$, where $\bar{V} \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^{k-1})$, we have $\|Vz\| \ge \|z\| \forall z \in \mathbb{R}^m$, whence the smallest singular value σ_{min} of V satisfies $\sigma_{min} \ge 1$. Since $p_k = u_k^{(k)}$ in Algorithm (12), we see from (2.1.54k) that $p_k^T w_k = u_k^{(k)T} w_k$ and $u_j^{(k)T} w_k = \omega_j^{(k)}$, $k \le j \le n$. By (12g), $|u_k^{(k)T} w_k| = \max\{|u_j^{(k)T} w_k| : k \le j \le n\}$. Lemma (26) therefore implies $\dfrac{|p_k^T w_k|}{\|p_k\|\|w_k\|} \ge \dfrac{1}{\|u_k^{(k)}\| \sqrt{n-k+1}}$, which with (32c) implies

$$\text{(33)} \qquad \frac{|p_k^T w_k|}{\|p_k\|\|w_k\|} \ge \sqrt{\frac{3}{(n-k+1)(2+4^{k-1})}} \equiv \eta_1^{(k)}.$$

Note that p_n and w_n lie in the same one-dimensional subspace, i.e. are multiples of one another, so LHS(33) = 1 for $k = n$. Thus (18) holds with $\eta_1 = \eta_1^{(\bar{k})}$, where $\eta_1^{(\bar{k})} = \min\{\eta_1^{(k)} \mid 1 \le k \le n-1\}$. Let $\psi(k) = (n-k+1)(2+4^{k-1})$, $1 \le k \le n-1$: then $\psi(\bar{k}) = \max \psi(\{1,2,\ldots,n-1\})$. Since

$$\psi(k+1) - \psi(k) = -2 + 4^{k-1}[4(n-k)-(n-k+1)]$$

$$= -2 + 4^{k-1}[3(n-k)-1]$$

is positive for $1 \le k \le n-2$, $\bar{k} = n-1$ and (18) holds with $\eta_1 = \eta_1^{(n-1)} = \text{RHS}(31)$. ∎

Now we can finally prove the fast local convergence of Brown's method.

(34) Theorem: Assume $f \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ are such that $f(x^*) = 0$ and $f'(x^*)$ is nonsingular. Suppose there exist $\mu \in [0,1]$, $\kappa \in [0,\infty)$, and $\rho_2 \in (0,\infty]$ such that

(35) $$\|f'(x) - f'(y)\| \le \kappa\|x-y\|^{\mu} \quad \forall\, x,y \in B(x^*,\rho_2).$$

Then for each $\gamma \in (0,\infty)$ there exists $\rho \in (0,\infty]$ such that the sequence $x_0, x_1, \ldots$ of major iterates produced by Algorithm (12) converges to x^* with Q-convergence rate $\sigma_Q(x_0,x_1,\ldots) \ge 1+\mu$ whenever $x_0 \in B(x^*,\rho)$. If $\mu = 0$ then the convergence is superlinear, and if $\mu > 0$ then the convergence rate $1+\mu$ is strong in the sense of (2.1.66).

Proof: First we show that the inner loop of Algorithm (12) satisfies the conditions of the inner loop of Algorithm (2.1.54) with $\xi = \kappa h^{\mu}$;

then we construct a Ξ: $\mathbb{R}^n \to [0,\infty]$ which makes Algorithm (12) a particular version of Algorithm (2.1.54), so that Theorem (2.1.62) applies.

Notice that if $\mu = 0$, then the continuity of f' implies that we may choose any $\kappa > 0$ and be assured that a $\rho_2 > 0$ exists such that (35) holds.

As noted above, Algorithm (12) chooses $p_k = u_k^{(k)}$ and $\pi_k = \omega_k^{(k)}$. By (12i) and (2.1.54k), $w_k^T u_j^{(k+1)} = 0$, $k+1 \leq j \leq n$, so we see inductively that $\mathrm{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\} = \mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$, $1 \leq k \leq n$. Hence $P_{k-1}p_k = P_{k-1}u_k^{(k)} = u_k^{(k)} = p_k$, $1 \leq k \leq n$, and (2.1.54f(i)) holds with $\delta = 0$.

Since we assume exact arithmetic, (2.1.54h) holds with $\epsilon_2 = 0$.

From (12f), (1.2.13), and (32c) we see that (2.1.54j) holds with $\epsilon_4\xi = [(2+4^{n-1})/3]^{(1+\mu)/2}\kappa h^{\mu}$, provided y_{k-1}, $y_{k-1} + hu_j^{(k)} \in B(x^*,\rho_2)$. Moreover, since $\|p_k\| = \|u_k^{(k)}\| \geq 1$, (2.1.54g) holds with $\epsilon_1 = \epsilon_4$, again provided y_{k-1}, $y_{k-1} + hu_k^{(k)} \in B(x^*,\rho_2)$.

As noted above, $\|U_k z\| \geq \|z\|$ $\forall$ $z \in \mathbb{R}^{n-k+1}$, so the least singular value $\sigma_{\min}$ of U_k satisfies $\sigma_{\min} = \|U_k\|/\chi(U_k) \geq 1$, whence (2.1.54i) holds with $c_0 = 1$.

Because $f(x^*) = 0$ and f is continuous, (12c) and (32c) imply the existence of $\rho_3 > 0$ such that $\rho_3 + h\|u_j^{(k)}\| \leq \rho_2$ whenever $y_0 = x_{i-1} \in B(x^*,\rho_3)$. Thus (2.1.54g) and (2.1.54j) hold as described above whenever $y_0, y_{k-1} \in B(x^*,\rho_3)$. In view of Lemma (30), Lemma (17) now implies the existence of $\rho_4 \in (0,\rho_3]$ and $\xi_1 > 0$ such that (2.1.54f(ii)) holds with $\eta = 2^{-n}$ and $\xi = \kappa h^{\mu}$ whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_4)$ and

$\kappa h^{\mu} \leq \xi_1$. The inner loop of Algorithm (12) thus meets the conditions of the inner loop of Algorithm (2.1.54) whenever the minor iterates lie in $B(x^*,\rho_4)$ and $\xi = \kappa h^{\mu} \leq \xi_1$.

From (2.1.22,24) and step (12c) we see that there exist $K \in (1,\infty)$ and $\rho_5 \in (0,\rho_4)$ such that $y_0,\ldots,y_n \in B(x^*,\min\{\rho_4, K\|y_0-x^*\|\})$ whenever $y_0 \in B(x^*,\rho_5)$ and $\kappa h^{\mu} \leq \kappa\gamma^{\mu}\|f(y_0)\|^{\mu} \leq \xi_1$. If $\mu = 0$, let ρ_4,ρ_5, and K be constructed for a particular choice of ρ_2,κ with K large enough that if $y_0 \in B(x^*,\rho_5)$ and $\kappa \leq \xi_1$ then $y_{k-1} + hu_j^{(k)} \in B(x^*,\min\{\rho_2, K\|y_0-x^*\|\})$. For $\mu = 0$ and $y_0 \in B(x^*,\rho_5)$, it suffices to assume $\kappa h^{\mu} = \kappa = \max\{\|f'(x)-f'(z)\| : x,z \in \bar{B}(x^*,K\|y_0-x^*\|)\}$, provided this value is $\leq \xi_1$; by the continuity of f', we may assume ρ_5 to be small enough that this is the case for all $y_0 \in B(x^*,\rho_5)$. Similarly, for $\mu > 0$ we may assume ρ_5 to be small enough that $\kappa h^{\mu} \leq \kappa\gamma^{\mu}\|f(y_0)\|^{\mu} \leq \xi_1 \ \forall\ y_0 \in B(x^*,\rho_5)$.

Let $\Xi: \mathbb{R}^n \to [0,\infty]$ be defined by

$$\Xi(x) = \begin{cases} \kappa\gamma^{\mu}\|f(x)\|^{\mu}, \text{ if } x \in B(x^*,\rho_5) \text{ and } \mu > 0; \\ \max\{\|f'(y)-f'(z)\| : y,z \in \bar{B}(x^*,K\|x-x^*\|)\}, \text{ if} \\ \qquad\qquad x \in B(x^*,\rho_5) \text{ and } \mu = 0; \\ +\infty, \text{ if } x \in \mathbb{R}^n - B(x^*,\rho_5). \end{cases}$$

We have just seen that the inner loop of Algorithm (12) meets the conditions of Algorithm (2.1.54) with $\xi = \Xi(y_0)$ whenever $y_0 = x_{i-1} \in B(x^*,\rho_5)$. Since $\Xi(y_0) = +\infty$ whenever $y_0 = x_{i-1} \in \mathbb{R}^n - B(x^*,\rho_5)$, Algorithm (2.1.54) allows arbitrary behavior outside $B(x^*,\rho_5)$ with this choice of Ξ, so this choice makes Algorithm (12) a particular case of Algorithm (2.1.54). Since $\Xi(x^*) = \lim_{\|x-x^*\|\to 0} \Xi(x) = 0$ for $\mu \geq 0$ and $\Xi(x) = (\kappa\gamma^{\mu})\|f(x)\|^{\mu}\ \forall x \in B(x^*,\rho_5)$ for $\mu > 0$, the assertions of (34) follow from Theorem (2.1.62). ∎

We conclude this subsection with two remarks. First, the value of η_1 given in (31) is the largest possible such that (18) holds for Algorithm (12). For example, if (12) is applied to $f(x) = Ax - b$ with

$$A = \begin{bmatrix} 1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & -1 & \cdots & -1 \\ \vdots & \ddots & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix},$$

then $\omega_k^{(k)} = 1$ and $\omega_j^{(k)} = -1$ for $j > k$, so no subscript interchanges are necessary in (12g), $1 \leq k \leq n$. Thus (12i) implies that (without interchanges) $u_j^{(k)} = \begin{pmatrix} -(2^{k-2}, 2^{k-3}, \ldots, 2, 1)^T \\ \cdots\cdots\cdots \\ e_{j-k+1} \end{pmatrix}$, so

$w_{n-1} = u_{n-1}^{(n-1)} - u_n^{(n-1)} = (0, \ldots, 0, 1, -1)^T$, and $p_{n-1} = u_{n-1}^{(n-1)}$ has

$$\frac{|w_{n-1}^T p_{n-1}|}{\|w_{n-1}\| \|p_{n-1}\|} = \frac{1}{\sqrt{2}\, \|p_{n-1}\|} = \frac{1}{\sqrt{2}\, \sqrt{1+(4^{n-2}-1)/3}}$$

$$= \sqrt{\frac{3}{2(2+4^{n-2})}}$$

$$= \mathrm{RHS}(31).$$

Second, (1.2.13) shows that the truncation error in $\frac{\Delta f_k}{\Delta u}(y,h)$ is bounded by a constant times $h^{\mu}\|u\|^{1+\mu}$. Therefore Algorithm (12) should suffer less truncation error if $\frac{\Delta f_k}{\Delta u_j^{(k)}}(u_{k-1},h)$ is replaced in step (12f) by $\|u_j^{(k)}\| \frac{\Delta f_k}{\Delta(u_j^{(k)}/\|u_j^{(k)}\|)}(y_{k-1},h)$, which also approximates $\frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1}) \equiv f_k'(y_{k-1})u_j^{(k)}$. The only effect of this change on Theorem (34) is to reduce ϵ_4 and ϵ_1 to 1. In practice, when finite precision arithmetic is used, it is crucial that the step size $h\|u\|$ used in computing $\frac{\Delta f_k}{\Delta u}(y,h)$ be neither too large nor too small, lest truncation or roundoff errors respectively become too severe; the suggested change to step (12f) allows precise control of this step size. Of course, the growth in $\|u_j^{(k)}\|$ which occurs in most practical problems will probably be considerably less spectacular than that in the example above. For such problems, we might expect the suggested change in step (12f) to cause only a minor improvement in the performance of (12).

2.3 Brown's Method with Finite Precision Arithmetic

Here we discuss implementing Brown's method in finite but variable precision arithmetic without reducing the rate of convergence. We conclude that in the practical case when only two fixed arithmetic precisions are available and a highly accurate solution is desired, it should suffice to do the housekeeping work of maintaining $u_k^{(k)},\ldots,u_n^{(k)}$ in single precision while performing the remaining calculations in double precision. We also propose a way to choose h in this case.

When finite difference methods like those considered here are carried out with exact arithmetic, truncation error (i.e. nonlinearity in the problem) alone prevents immediate convergence. In practice, however, such methods are implemented on computers with only finite precision arithmetic, so that roundoff error too may hinder convergence. When the truncation error is large, high-precision arithmetic takes more effort than lower-precision arithmetic while yielding little improvement in performance. Thus, from the standpoint of computational effort, it appears best to compute only to the arithmetic accuracy warranted by the probable truncation error.

Although many present computers offer only two arithmetic precisions in floating point, namely "single" and "double" precision, it is conceivable that some future machines will boast efficient variable-precision floating-point arithmetic. The methods considered here may be implemented using finite but variable precision arithmetic without suffering a reduction in their rate of Q-convergence. By way of illustration, we will see in this subsection how to implement Brown's method in finite but variable precision arithmetic while retaining the rate of convergence described in Theorem (2.2.34).

We shall show that the following finite-precision form of Algorithm (2.2.12) behaves like (2.2.12).

(1) Algorithm: [Variable precision floating-point version of Brown's method];

(1a) Let $f \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, γ, δ_1, $\delta_2 \in (0,\infty)$, $\bar{\mu} \in [0,1]$ be given;

OUTERLOOP:

DO $i = 1$ TO ∞;

(1b) If $f(x_{i-1}) = 0$ then stop and let $x_j = x_{i-1}$ for all $j \geq i$;

(1c) Choose $h \in (0, \gamma \|f(x_{i-1})\|]$;

Choose $\lambda_1, \lambda_2 > 0$ with

(1d) $$\lambda_1 \leq \delta_1 h^{\bar{\mu}} \quad \text{and}$$

(1e) $$\lambda_2 \leq \delta_2 h^{1+\bar{\mu}} / \max\{1, \|x_{i-1}\|\}$$

and let $fl_j(\cdot)$ denote the value obtained when $(\cdot)$ is evaluated in floating-point arithmetic with unit roundoff λ_j, $j = 1,2$;

(1f) $y_0 \leftarrow x_{i-1}$;

(1g) Let $\tilde{u}_j^{(1)} = e_j, \quad 1 \leq j \leq n$;

INNERLOOP:

DO $k = 1$ TO n;

(1h) $$\omega_j^{(k)} \leftarrow fl_2\left(\frac{\Delta f_k}{\Delta u_j^{(k)}} (y_{k-1}, h)\right) \quad \text{for } k \leq j \leq n;$$

(1i) Relabel if necessary so that

$$|\omega_k^{(k)}| = \max\{|\omega_j^{(k)}|: \ k \leq j \leq n\};$$

(1j) $$y_k \leftarrow fl_2\left(y_{k-1} - \left[\frac{f_k(y_{k-1})}{\omega_k^{(k)}}\right] \tilde{u}_k^{(k)}\right);$$

For $k < j \leq n$, either

(1k) $$\tilde{u}_j^{(k+1)} \leftarrow fl_1\left(\tilde{u}_j^{(k)} - \left[\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right] \tilde{u}_k^{(k)}\right) \quad \text{or}$$

(1k)' $\tilde{u}_j^{(k+1)} \leftarrow fl_1(\Omega_{k+1}^{-1} e_j)$ (by backwards substitution), with Ω_{k+1} as in (2.2.9);

END INNERLOOP;

$x_i \leftarrow y_n$;

END OUTERLOOP and Algorithm (i);

We assume that it is possible to compute a lower bound on $\|f(x_{i-1})\|$, so that h and the arithmetic precisions $fl_1(\cdot)$ and $fl_2(\cdot)$ may be determined at the start of each major iteration and the stopping test (1b) may be carried out.

For small h, $fl_2(\cdot)$ computes to roughly twice as many bits of precision as $fl_1(\cdot)$ in the usual case when $\bar{\mu} = 1$. In this sense, $fl_1(\cdot)$ does single-precision arithmetic, whereas $fl_2(\cdot)$ does double-precision. "Double" precision is needed in (1h) because of the division by h in $\frac{\Delta f_k}{\Delta u}(y,h)$. It is also needed in (1j) to assure accurate determination of the minor iterates. As we shall soon see, the accuracy of the $\tilde{u}_j^{(k)}$ is less critical.

Suppose Algorithm (1) is used go find a sequence $x_1, x_2, \ldots$ of better and better approximations to a nonsingular zero x^* of $f \in C^1(\mathbb{R}^n)$, starting from a sufficiently good initial guess x_0. Suppose further that for some $\mu \in (0,1]$ f' satisfies a Lipschitz condition of order μ near x^*, i.e. there exist $\kappa \in [0,\infty)$ and $\rho \in (0,\infty]$ such that $\|f'(x)-f'(y)\| \leq \kappa\|x-y\|^{\mu}$ for all $x \in B(x^*,\rho)$. It is intended in this case that the constant $\bar{\mu}$ used in steps (1d) and (1e) be close to μ, since $O_Q(x_0,x_1,\ldots) \geq 1 + \min\{\mu,\bar{\mu}\}$. If $\bar{\mu}$ is too small, then the convergence rate will be lower than it could be, while if $\bar{\mu}$ is too large, then computational effort will be wasted on superfluous arithmetic precision. Of course, in most cases of practical interest f' will satisfy a Lipschitz condition of order 1, so that choosing $\bar{\mu} = 1$ will not waste effort on needless accuracy. Note that if $\mu > \bar{\mu}$, then f' also satisfies a Lipschitz condition of order $\bar{\mu}$ near x^*; in the results proved below we thus assume WLOG that $\mu \leq \bar{\mu}$.

Brown and Dennis (1971) have (tacitly) noted that the division by h in step (1h) may be omitted, provided we take this omission into account in step (1j): step (1k) or (1k)' is unaffected if both $\omega_j^{(k)}$ and $\omega_k^{(k)}$ are scaled by h, except that the results may be more accurate under the assumed finite precision arithmetic.

Assume the arithmetic used in (1) to be such that $|fl(s/t)| \leq 1$ whenever $0 < |t| \geq |s|$.

We now prove several lemmas which enable us to show that Algorithm (1) meets the conditions of Algorithm (2.1.54) near a nonsingular zero x^* of f. To do this, we adopt the view that Algorithm (1) implicitly generates $u_j^{(k)}$ with exact arithmetic by

(2a) $$u_j^{(1)} = e_j, \quad 1 \leq j \leq n, \quad \text{and}$$

(2b) $$u_j^{(k+1)} = u_j^{(k)} - \left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}} \right) u_k^{(k)}, \quad k \leq j \leq n.$$

According to this view, step (1h) computes $\omega_j^{(k)}$ as an approximation to $\frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})$, but chooses $p_k = \tilde{u}_k^{(k)}$ while $\pi_k = \omega_k^{(k)}$.

First we establish bounds on $\|\tilde{u}_j^{(k)}\|$ and $\|\tilde{u}_j^{(k)} - u_j^{(k)}\|$.

(3) <u>**Lemma**</u>: For each $n \geq 1$ and each sufficiently small $\bar{\lambda}_1 \in (0,\infty)$ there exist constants $K_2, K_3 \in (0,\infty)$ such that

(4) $$\|\tilde{u}_j^{(k)} - u_j^{(k)}\| \leq K_2 \lambda_1 \quad \text{and}$$

(5) $$\|\tilde{u}_j^{(k)}\| \leq K_3, \qquad 1 \leq k \leq n, \quad k \leq j \leq n$$

whenever $\tilde{u}_j^{(k)}$ are generated by Algorithm (1) with $\lambda_1 \leq \bar{\lambda}_1$ and $u_j^{(k)}$ are generated by (2) using the $\omega_j^{(k)}$ from Algorithm (1).

<u>Proof</u>: It suffices to consider separately the cases when (1k) and (1k)' respectively are used. WLOG no relabeling occurs in step (1i).

Assume first that (1k) is used. Let $\tilde{u}_{j,\nu}^{(k)} = e_\nu^T \tilde{u}_j^{(k)}$ denote the ν^{th} component of $\tilde{u}_j^{(k)}$. Since $|fl(\omega_j^{(k)}/\omega_k^{(k)})| \leq 1$, we have $|\tilde{u}_{j,\nu}^{(k+1)}| \leq (1+\lambda_1)(|\tilde{u}_{j,\nu}^{(k)}| + |\tilde{u}_{k,\nu}^{(k)}|)$, so by induction we find that

(6a) $$|\tilde{u}_{j,\nu}^{(k)}| \leq (2+2\bar{\lambda}_1)^{k-1-\nu}, \qquad 1 \leq \nu \leq k-1,$$

while

(6b) $$\tilde{u}_{j,j}^{(k)} = 1 \quad \text{and} \quad \tilde{u}_{j,\nu}^{(k)} = 0 \quad \text{for} \quad \nu \leq k, \ \nu \neq j,$$

whenever $\lambda_1 \leq \bar{\lambda}_1$. Thus (5) holds with $K_3 = \sqrt{1 + (2+2\bar{\lambda}_1)^{2n-2}/(3+8\bar{\lambda}_1+4\bar{\lambda}_1^2)}$.

Consider now a particular component assignment $\tilde{u}_{j,\nu}^{(k+1)} \leftarrow fl_1\left(\tilde{u}_{j,\nu}^{(k)} - \left[\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right]\tilde{u}_{k,\nu}^{(k)}\right)$.

Since $\lambda_1 \leq \bar{\lambda}_1$ is the unit roundoff, there exist $\lambda_1^{(1)}, \lambda_1^{(2)}, \lambda_1^{(3)} \in [-\lambda_1, \lambda_1]$ such that $\tilde{u}_{j,\nu}^{(k+1)} = (1+\lambda_1^{(3)})\left[\tilde{u}_{j,\nu}^{(k)} - (1+\lambda_1^{(2)})(1+\lambda_1^{(1)})\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)\tilde{u}_{k,\nu}^{(k)}\right]$. Letting

$u_{j,\nu}^{(k)} = e_\nu^T u_j^{(k)}$ denote the $\nu^{\underline{th}}$ component of $u_j^{(k)}$, we thus have

$$|\tilde{u}_{j,\nu}^{(k+1)} - u_{j,\nu}^{(k+1)}| \le |\tilde{u}_{j,\nu}^{(k)} - u_{j,\nu}^{(k)}| + \left|\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)\left(\tilde{u}_{k,\nu}^{(k)} - u_{k,\nu}^{(k)}\right)\right| +$$

$$+ \left|[(1+r_1^{(2)})(1+r_1^{(1)}) - 1]\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\right)\tilde{u}_{k,\nu}^{(k)}\right| + \left|r_1^{(3)}\left[\tilde{u}_{j,\nu}^{(k)} - fl_1\left(\frac{\omega_j^{(k)}}{\omega_k^{(k)}}\,\tilde{u}_{k,\nu}^{(k)}\right)\right]\right|.$$

In view of (6a) and the inequalities $|\omega_j^{(k)}/\omega_k^{(k)}| \le 1$, $|fl_1(\omega_j^{(k)}/\omega_k^{(k)})| \le 1$, and $|r_1^{(1)}|$, $|r_1^{(2)}|$, $|r_1^{(3)}| \le \lambda_1$, we thus find for $\bar{\lambda}_1 \le 1$ that

$$|\tilde{u}_{j,\nu}^{(k+1)} - u_{j,\nu}^{(k+1)}| \le |\tilde{u}_{j,\nu}^{(k)} - u_{j,\nu}^{(k)}| + |\tilde{u}_{k,\nu}^{(k)} - u_{k,\nu}^{(k)}| + 5\lambda_1(2+2\bar{\lambda}_1)^{k-1-\nu}$$

for $1 \le \nu \le k-1$. Since $|\tilde{u}_{j,k}^{(k+1)} - u_{j,k}^{(k+1)}| \le \lambda_1$ while $\tilde{u}_{j,\nu}^{(k+1)} = u_{j,\nu}^{(k+1)}$ for $\nu \ge k+1$, we see that if

(7a) $\qquad \zeta_\ell = 0$ for $\ell \le 0$,

(7b) $\qquad \zeta_1 = 1$, and

(7c) $\qquad \zeta_{\ell+1} = 2\zeta_\ell + 5(2+2\bar{\lambda}_1)^{\ell-1} = 2^\ell\left(1+\frac{5}{2}\sum_{i=0}^{\ell-1}\left(1+\bar{\lambda}_1\right)^i\right)$ for $\ell \ge 1$,

then $|\tilde{u}_{j,\nu}^{(k)} - u_{j,\nu}^{(k)}| \le \zeta_{k-\nu}\lambda_1$. Hence (4) holds with

(8) $$K_2 = \left(\sum_{\ell=1}^{n-1} \zeta_\ell^2\right)^{1/2}.$$

Assume now that (1k)' is used. Following Forsythe and Moler (1967, pp. 104-105), we find $K_4 \in (0,\infty)$ such that for each j,k with $1 \le k \le j \le n$ there exists $E \in \mathcal{L}(\mathbb{R}^n)$ with

(9) $$(\Omega_k + E)\tilde{u}_j^{(k)} = e_j \quad \text{and}$$

$$\|E\| \le K_4 \lambda_1 ,$$

provided $n\lambda_1 \le .01$. Since Ω_k is upper triangular with ones on the diagonal and all off-diagonal entries $\omega_j^{(k)}/\omega_k^{(k)}$ bounded by 1 in absolute value, there exists $K_5 \in (0,\infty)$ such that $\|\Omega_k^{-1}\| \le K_5$ for any such Ω_k. Assume $\bar{\lambda}_1 K_4 K_5 < 1$, which defines a "sufficiently small" $\bar{\lambda}_1$ in the hypotheses. Since

$$\|(\Omega_k + E)^{-1}\| \le \frac{\|\Omega_k^{-1}\|}{1 - \|E\|\|\Omega_k^{-1}\|} \le \frac{K_5}{1-\bar{\lambda}_1 K_4 K_5} ,$$

(9) implies that (5) holds with $K_3 = K_5/(1 - \bar{\lambda}_1 K_4 K_5)$. Moreover, since $\Omega_k u_j^{(k)} = e_j$, we have

$$\begin{aligned} \tilde{u}_j^{(k)} - u_j^{(k)} &= [(\Omega_k + E)^{-1} - \Omega_k^{-1}]e_j \\ &= [(\Omega_k + E)^{-1}(\Omega_k - [\Omega_k + E])\Omega_k^{-1}]e_j, \end{aligned}$$

whence
$$\begin{aligned} \|\tilde{u}_j^{(k)} - u_j^{(k)}\| &\le \|(\Omega_k + E)^{-1}\|\|E\|\|\Omega_k^{-1}\| \\ &\le \left(\frac{K_4 K_5^2}{1-\bar{\lambda}_1 K_4 K_5}\right)\lambda_1, \end{aligned}$$

so (4) holds with $K_2 = K_4 K_5^2/(1-\bar{\lambda}_1 K_4 K_5)$. ∎

For small $\bar{\lambda}_1$, the K_2 given by (7) and (8) for (1k) is roughly $2^{n-3}(3n)$. On the other hand, $K_4 > 1$ and $\|\Omega_n^{-1}\|$ can be larger than

$2^n/\sqrt{12}$, whence $K_5 > 2^n/\sqrt{12}$, so the K_2 just derived for (1k)' is larger than the one for (1k) by an exponentially growing factor. However, I doubt whether the former bound is sharp.

Lemma (2.2.30) combines with (4) to yield an η_1 and ρ_3 for which (2.2.18) holds:

(10) Lemma: Suppose $f \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ satisfy $f(x^*) = 0$ with $f'(x^*)$ nonsingular. Assume that the parameters $\gamma, \delta_1, \delta_2 \in (0,\infty)$ and $\bar{\mu} \in (0,1]$ of Algorithm (1) are fixed. Then for any $\theta \in (0,1)$ there exists $\rho_3 \in (0,\infty)$ such that (2.2.18) holds with

(11) $$\eta_1 = \theta \sqrt{\frac{3}{2(2+4^{n-2})}}$$

whenever

$$y_0 \in B(x^*, \rho_3).$$

Proof: Fix k and let $u_j^{(k)}$ be generated by (2). For the moment, let $u = u_k^{(k)}$ and $\tilde{u} = \tilde{u}_k^{(k)}$. The proof of Lemma (2.2.30) shows that

$|u^T w_k| \geq \bar{\eta}_1 \|u\| \|w_k\|$, where $\bar{\eta}_1 \equiv \mathrm{RHS}(2.2.31) = \sqrt{\frac{3}{2(2+4^{n-2})}}$. Let $\bar{\lambda}_1 \in (0,1)$ satisfy the "sufficiently small" condition of Lemma (3), and let K_2 be such that (4) holds. If $\lambda_1 \leq \bar{\lambda}_1$, we then have:

$$\begin{aligned} |\tilde{u}^T w_k| &\geq |u^T w_k| - \|u - \tilde{u}\| \|w_k\| \\ &\geq (\bar{\eta}_1 \|u\| - \|u - \tilde{u}\|) \|w_k\| \\ &\geq (\bar{\eta}_1 \|\tilde{u}\| - (1+\bar{\eta}_1) K_2 \lambda_1) \|w_k\|, \quad \text{since } \|u\| \geq \|\tilde{u}\| - \|u - \tilde{u}\| \end{aligned}$$

$$\geq (\bar{\eta}_1 - (1+\bar{\eta}_1)K_2 \mathit{h}_1)\|\tilde{u}\|\|w_k\|, \quad \text{since } \|\tilde{u}\| \geq 1.$$

Since $p_k = \tilde{u}_k^{(k)} = \tilde{u}$, (2.2.18) holds with η_1 given by (11), i.e. $|p_k^T w_k| \geq \theta\bar{\eta}_1\|p_k\|\|w_k\|$, whenever $(1+\bar{\eta}_1)K_2\mathit{h}_1 \leq (1-\theta)\bar{\eta}_1$. Because of (1c) and (1d), this is guaranteed to happen whenever $y_0 \in B(x^*,\rho_3)$, where $\rho_3 \in (0,\infty)$ is so small that

$$\max\|f(\bar{B}(x^*,\rho_3))\| \leq \gamma^{-1}\left[\frac{(1-\theta)\bar{\eta}_1}{\delta_1(1+\bar{\eta}_1)K_2}\right]^{1/\bar{\mu}} ;$$

since f is continuous with $f(x^*) = 0$, such a ρ_3 exists. ∎

Next we bound $\left|\frac{\Delta f_k}{\Delta u}(y,h) - \frac{\partial f_k}{\partial u}(y)\right|$.

(12) <u>Lemma</u>: Let $f \in C^1(\mathbb{R}^n)$ and let $\bar{\mu} \in (0,1]$ and $\delta_2 \in (0,\infty)$ be given. Suppose for some $x^* \in \mathbb{R}^n$, $\mu \in [0,\bar{\mu}]$, $\kappa \in [0,\infty)$, and $\rho_2 \in (0,\infty]$ that

(13) $$\|f'(x) - f'(y)\| \leq \kappa\|x-y\|^{\mu} \quad \text{for all } x,y \in B(x^*,\rho_2).$$

Then there exists $\kappa_1 \in (0,\infty)$ such that for any $h \in (0,1)$ and $u \in \mathbb{R}^n$,

(14) $$\left|fl_2\left(\frac{\Delta f_k}{\Delta u}(y,h)\right) - \frac{\partial f_k}{\partial u}(y)\right| \leq (\kappa + h^{\bar{\mu}-\mu}\kappa_1)h^{u} \max\{1,\|u\|\}$$

whenever $y, y+hu \in B(x^*,\rho_2)$, $1 \leq k \leq n$, and $fl_2(\cdot)$ denotes the result of evaluating $(\cdot)$ in floating-point arithmetic with unit roundoff h_2 satisfying

(15) $$\mathit{h}_2 \leq \delta_2 h^{1+\bar{\mu}}.$$

<u>Proof</u>: Let $\tilde{f}_k \equiv fl_2(f_k)$ denote the floating-point approximation to f_k used in computing $fl_2\left(\frac{\Delta f_k}{\Delta u}(y,h)\right)$; thus, for any $\mathbf{x} \in \mathbb{R}^n$ we have $|\tilde{f}_k(x) - f_k(x)| \le \kappa_2 |f_k(x)|$. Let

(16a) $$K_0 = \max\|f(\bar{B}(x^*,\rho_2))\| \quad \text{and}$$

(16b) $$K_1 = \max\|f'(\bar{B},x^*,\rho_2))\|.$$

Given y,h,u with $y, y+hu \in B(x^*,\rho_2)$ and $h \in (0,1)$, there exist $\kappa_2^{(1)}, \kappa_2^{(2)} \in [-\kappa_2, \kappa_2]$ such that

$$fl_2\left(\frac{\Delta f_k}{\Delta u}(y,h)\right) = (1+\kappa_2^{(1)})(1+\kappa_2^{(2)})\frac{\tilde{f}_k(y+hu)-\tilde{f}_k(y)}{h}.$$

Thus

$$\begin{aligned} h[\mathrm{LHS}(14)] &= |(\tilde{f}_k(y+hu) - \tilde{f}_k(y))(1+\kappa_2^{(1)})(1+\kappa_2^{(2)}) - hf'(y)u| \\ &\le (1+\kappa_2)^2[|(\tilde{f}_k-f_k)(y+hu)| + |(\tilde{f}_k-f_k)(y)|] + \\ &\quad + \kappa_2(2+\kappa_2)|f_k(y+hu) - f_k(y)| + \\ &\quad + |f_k(y+hu) - f_k(y) - hf'(y)u|. \end{aligned}$$

By the MVT there exists $\theta \in (0,1)$ such that $f_k(y+hu) - f_k(y) = hf_k'(y+\theta hu)u$, so, using (16) and (13), we find

$$h[\mathrm{LHS}(14)] \le 2(1+\kappa_2)^2 K_0\kappa_2 + hK_1\|u\|\kappa_2(2+\kappa_2) + \kappa h^{1+\mu}\|u\|^{\mu}.$$

Hence (15) and the hypotheses $h \in (0,1)$ and $\mu \in [0,\bar{\mu}]$ yield

$$\text{LHS}(14) \le 2(1+\delta_2)^2K_0\delta_2h^{\bar{\mu}} + K_1\|u\|\delta_2(2+\delta_2)h^{1+\bar{\mu}} + \kappa h^{\mu}\|u\|^{\mu}$$

$$\le \left([2(1+\delta_2)K_0+K_1]\delta_2(2+\delta_2)h^{\bar{\mu}-\mu} + \kappa\right) h^{\mu}\max\{1,\|u\|\},$$

so (14) holds with $\kappa_1 = [2(1+\delta_2)K_0+K_1]\delta_2(2+\delta_2)$. ∎

Lemma (12) enables us to show that Algorithm (1) meets conditions (2.1.54g,j). The next lemma will give us a δ for which (2.1.54f(i)) holds.

(17) Lemma: For each $n \ge 1$ and each $\bar{\lambda}_1 \in (0,1)$ small enough that Lemma (3) holds, there exists K_4 such that

(18) $$\|\tilde{U}_k\tilde{U}_k^+ - U_kU_k^+\| \le K_4\lambda_1$$

whenever Algorithm (1) runs with $\lambda_1 \le \bar{\lambda}_1$, where

(19a) $$\tilde{U}_k = [\tilde{u}_k^{(k)},\ldots,\tilde{u}_n^{(k)}] \text{ and}$$

(19b) $$U_k = [u_k^{(k)},\ldots,u_n^{(k)}],$$ as in (2.1.52), with $u_j^{(k)}$ generated by (2).

Proof: WLOG no relabeling occurs in step (1i). Both $\tilde{U}_k$ and U_k then have the form $\begin{pmatrix} X \\ I_{n-k+1}\end{pmatrix}$, where $X \in \mathcal{L}(\mathbb{R}^{n-k+1}, \mathbb{R}^{k-1})$, so $\|\tilde{U}_kz\| \ge \|z\|$ and $\|U_kz\| \ge \|z\|$ for all $z \in \mathbb{R}^{n-k+1}$, whence all singular values of $\tilde{U}_k$ and U_k are ≥ 1. From the singular values decomposition theorem we thus have

(20a) $$\|(\tilde{U}_k^T\tilde{U}_k)^{-1}\tilde{U}_k^T\| \le 1 \quad \text{and}$$

(20b) $$\|U_k(U_k^TU_k)^{-1}\| \le 1.$$

Since

$$\begin{aligned}\tilde{U}_k\tilde{U}_k^+ - U_kU_k^+ &= \tilde{U}_k(\tilde{U}_k^T\tilde{U}_k)^{-1}\tilde{U}_k^T - U_k(U_k^TU_k)^{-1}U_k^T \\ &= (\tilde{U}_k-U_k)[(\tilde{U}_k^T\tilde{U}_k)^{-1}\tilde{U}_k^T] + [U_k(U_k^TU_k)^{-1}][U_k^T(U_k-\tilde{U}_k) + \\ &\quad + (U_k-\tilde{U}_k)^T\tilde{U}_k][(\tilde{U}_k^T\tilde{U}_k)^{-1}\tilde{U}_k^T] + [U_k(U_k^TU_k)^{-1}](\tilde{U}_k-U_k)^T,\end{aligned}$$

we have by (20) that

$$\|\tilde{U}_k\tilde{U}_k^+ - U_kU_k^+\| \le (2 + \|U_k\| + \|\tilde{U}_k\|)\|\tilde{U}_k - U_k\|.$$

Combined with (4), (5), and (2.2.32c), this implies

$$\|\tilde{U}_k\tilde{U}_k^+ - U_kU_k^+\| \le \left(2 + \left[K_3 + \sqrt{(2+4^{k-1})/3}\right]\sqrt{n-k+1}\right)\sqrt{n-k+1}\; K_2 h_1,$$

whence (18) follows with $K_4 = \left(2\sqrt{n} + n[K_3+2^{n-1}]\right) K_2$. ∎

We now use the above lemmas to prove

(21) Theorem: Suppose $f \in C^1(\mathbb{R}^n)$ and $x^* \in \mathbb{R}^n$ satisfy $f(x^*) = 0$ with $f'(x^*)$ nonsingular. Let $\bar{\mu} \in (0,1]$ be fixed and assume there exist $\mu \in [0,\bar{\mu}]$, $\rho_2 \in (0,\infty]$, and $\kappa \in [0,\infty)$ such that (13) holds, i.e.

(22) $$\|f'(x) - f'(y)\| \le \kappa\|x-y\|^{\mu} \quad \text{for all } x,y \in B(x^*,\rho_2).$$

Then for each choice of the constants γ, $\delta_1, \delta_2 \in (0,\infty)$ there exists

$\rho \in (0,\infty)$ such that if $x_0 \in B(x^*,\rho)$, then the sequence $x_0, x_1, \ldots$ of major iterates generated from x_0 by Algorithm (1) converges to x^* with Q-order $\sigma_Q(x_0,x_1,\ldots) \geq 1+\mu$. If $\mu = 0$ then this convergence is superlinear, while if $\mu > 0$, then the convergence rate $1+\mu$ is strong in the sense of (2.1.66).

<u>Proof</u>: We first find $\rho_7, \delta, \epsilon_1, \epsilon_2, \epsilon_4 \in (0,\infty)$ and $\eta \in (0,1)$ such that the inner loop of Algorithm (1) meets the conditions of the inner loop of Algorithm (2.1.54) with $\xi = O(\kappa h^{\mu} + h^{\bar{\mu}})$, $p_k = \tilde{u}_k^{(k)}$, $\pi_k = \omega_k^{(k)}$, $u_j^{(k)}$ given by (2), and w_k as in (2.1.54k) whenever $y_0 = x_{i-1} \in B(x^*,\rho_7)$; then we construct $\Xi: \mathbb{R}^n \to [0,\infty]$ such that Algorithm (1) becomes a special case of Algorithm (2.1.54) and the assertions of (21) follow from Theorem (2.1.62).

Let K_0, K_1 be given by (16) and let $\bar{\lambda}_1$ be small enough that Lemma (3) holds. Let K_2, K_3 be given by Lemma (3). Because of the continuity of f and the assumptions $f(x^*) = 0$, $\gamma \in (0,\infty)$, and $\bar{\mu} \in (0,1]$, there exists $\rho_5 > 0$ such that h chosen in step (1c) and λ_1 chosen in (1d) satisfy $h \in (0,1)$, $\lambda_1 \leq \bar{\lambda}_1$, and $\rho_5 + K_3 h \leq \rho_2$ whenever $y_0 = x_{i-1} \in B(x^*,\rho_5)$. Thus if $y_0, y_{k-1} \in B(x^*,\rho_5)$, then $h \in (0,1)$, $\lambda_1 \leq \bar{\lambda}_1$, and y_{k-1}, $y_{k-1} + hu_j^{(k)} \in B(x^*,\rho_2)$.

Let $u_k^{(k)}, \ldots, u_n^{(k)}$ be generated by (2) and define $\tilde{U}_k, U_k$ as in (19). Since $\mathrm{Span}\{u_k^{(k)}, \ldots, u_n^{(k)}\} = \mathrm{Span}\{w_1, \ldots, w_{k-1}\}^{\perp}$, as shown in the proof of Theorem (2.2.34), where $w_j = U_j(U_j^T U_j)^{-1}\begin{pmatrix} \omega_j^{(j)} \\ \vdots \\ \omega_n^{(j)} \end{pmatrix}$ (as in (2.1.54k)), $P_{k-1} = U_k U_k^+$. But $p_k = \tilde{u}_k^{(k)}$ implies $p_k = \tilde{U}_k \tilde{U}_k^+ p_k$, so

$p_k - P_{k-1}p_k = (\tilde{U}_k\tilde{U}_k^+ - U_kU_k^+)p_k$. With K_4 from Lemma (17) we therefore have

(23) $$\|p_k - P_{k-1}p_k\| \le K_4 r_1\|p_k\| \le K_4\delta_1 h^{\bar{\mu}}\|p_k\|$$

whenever $y_0 = x_{i-1} \in B(x^*,\rho_5)$.

Since $p_k = \tilde{u}_k^{(k)}$ and $\pi_k = \omega_k^{(k)} = fl_2\left(\frac{\Delta f_k}{\Delta\tilde{u}_k^{(k)}}\right)(y_{k-1},h)$, Lemma (12), (5), and the inequality $K_3 \ge \|u_k^{(k)}\| \ge 1$ imply for some $\kappa_1 \in (0,\infty)$ that

(24) $$|\pi_k - \frac{\partial f_k}{\partial p_k}(y_{k-1})| \le (\kappa h^{\mu} + \kappa_1 h^{\bar{\mu}})K_3$$

and similarly, with help from (16b), (4), and (1d), that

(25) $$|\omega_j^{(k)} - \frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})| \le \kappa K_3 h^{\mu} + (\kappa_1 K_3 + \delta_1 K_1 K_2)h^{\bar{\mu}}$$

whenever $y_0, y_{k-1} \in B(x^*,\rho_5)$.

As in the proof of Theorem (2.2.34) we have $\chi(U_k) \le \|U_k\|$, so (2.1.54i) holds with $c_0 = 1$.

By Lemmas (10) and (2.2.17) there exist ξ_1, $\rho_4 > 0$ such that

(26) $$\left|\frac{\partial f_k}{\partial p_k}(y_{k-1})\right| \ge 2^{1-n}\|P_{k-1}\nabla f_k(y_{k-1})\|\|p_k\|$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\min\{\rho_4,\rho_5\})$ and

(27) $$\kappa K_3 h^{\mu} + (\kappa_1 K_3 + \delta_1 K_1 K_2)h^{\bar{\mu}} \le \xi_1.$$

For $\mu = 0$, let the foregoing have been carried out for a particular choice of κ,ρ_2 such that (22) holds. In all cases, the continuity of f,f' and choice of h in (1c) imply the existence of $\rho_6 \in (0,[\min \rho_4,\rho_5])$ such that (27) and hence (26) hold whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_6)$. For $\mu = 0$, it suffices to choose ρ_6 so that (27) holds with $h = \gamma \max\|f(\bar{B}(x^*,\rho_6))\|$ and $\kappa = \max\{\|f'(y)-f'(z)\|;$ $y,z \in \bar{B}(x\ ,\rho_6+K_3h\}$, since $y_j + hu_\ell^{(j)} \in B(x^*,\rho_6+K_3h)$ whenever $y_0,y_j \in B(x^*,\rho_6)$.

By Lemma (2.1.15) we may assume ρ_6 to be so small that

(28) $$\|P_{k-1}\nabla f_k(y_{k-1})\| \geq \frac{1}{2\|f'(x^*)^{-1}\|}$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_6)$. From (24), (26), and (28),

$$|\pi_k| \geq (2^n\|f'(x^*)^{-1}\|)^{-1} - (\kappa h^\mu+\kappa_1 h^{\bar{\mu}})K_3.$$

By the continuity and choice of h we may thus assume ρ_6 to be so small that

(29) $$|\pi_k| = |\omega_k^{(k)}| \geq 1/(2^{n+1}\|f'(x^*)^{-1}\|)$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_6)$.

At last we can estimate the accuracy of step (1j). Assume $y_0,\ldots,y_{k-1} \in B(x^*,\rho_6)$ and let $p_k^\nu = e_\nu^T p_k$, $y_{k-1}^\nu = e_\nu^T y_{k-1}$, and $y_k^\nu = e_\nu^T y_k$ denote the ν^{th} components of p_k,y_{k-1}, and y_k. Since λ_2 is the unit roundoff for fl_2, for each ν, $1 \leq \nu \leq n$, there exist $\lambda_2^{(j)} \in [-\lambda_2,\lambda_2]$, $j = 1,2,3,4$, such that $fl_2(f_k(y_{k-1})) = f_k(y_{k-1})(1+\lambda_2^{(1)})$ and

$$y_k^\nu = fl_2\left(y_{k-1}^\nu - \frac{f_k(y_{k-1})}{\pi_k} p_k^\nu\right)$$

$$= \left[y_{k-1}^\nu - \frac{f_k(y_{k-1})(1+\lambda_2^{(1)})}{\pi_k} p_k^\nu (1+\lambda_2^{(2)})(1+\lambda_2^{(3)})\right](1+\lambda_2^{(4)}).$$

Thus $\left| y_k^\nu - \left(y_{k-1}^\nu - \frac{f_k(y_{k-1})}{\pi_k} p_k^\nu\right)\right| \le \lambda_2 |y_{k-1}^\nu| + [(1+\lambda_2)^4 - 1]\left|\frac{f_k(y_{k-1})p_k^\nu}{\pi_k}\right|$.

WLOG ρ_6 is small enough that $6\lambda_2 + 4\lambda_2^2 + \lambda_2^3 \le 1$ whenever $y_0 \in B(x^*,\rho_6)$, whence $(1+\lambda_2)^4 - 1 \le 5\lambda_2$. Also, WLOG $\rho_6 \le 1/2$, so that $\|y_{k-1}\| \le \|y_0\| + 1$ whenever $y_0, y_{k-1} \in B(x^*,\rho_6)$. Using (1e), (1c), (16a), (5), and (29), we thus find

$$\left\| y_k - \left(y_{k-1} - \frac{f_k(y_{k-1})}{\pi_k} p_k\right)\right\| \le [2+5(2^{n+1})\|f'(x^*)^{-1}\|K_0K_3]\sqrt{n}\,\delta_2 \gamma h^{\bar{\mu}}\|f(y_0)\| \tag{30}$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho_6)$.

From (23), (24), (25), (26), (30), and (1c), we see that there exist δ, $\epsilon_1,\epsilon_2,\epsilon_4,K_5,K_6 \in (0,\infty)$ and $\eta = 2^{1-n}$ such that the inner loop of Algorithm (1) meets the conditions of the inner loop of Algorithm (2.1.54) with $c_0 = 1$ and $\xi = K_5\kappa\|f(y_0)\|^\mu + K_6\|f(y_0)\|^{\bar{\mu}}$ whenever $y_0,\ldots,y_n \in B(x^*,\rho_6)$. By continuity and Lemma (2.1.20), there exist $K,\rho_7 > 0$ such that $y_0,\ldots,y_n \in B(x^*,\min\{\rho_6,K\|y_0-x^*\|\})$ whenever $y_0 \in B(x^*,\rho_7)$. Let $\Xi\colon \mathbb{R}^n \to [0,\infty]$ be given by

$$\Xi(x) = \begin{cases} K_5\kappa\|f(x)\|^{\mu} + K_6\|f(x)\|^{\bar{\mu}} & \text{if } x \in B(x^*,\rho_7) \text{ and } \mu > 0 \\ K_5 \max\{\|f'(y)-f'(z)\|: y,z \in B(x^*,K\|x-x^*\|)\} + K_6\|f(x)\|^{\bar{\mu}} & \\ \qquad\qquad \text{if } x \in B(x^*,\rho_7) \text{ and } \mu = 0 \\ +\infty \text{ if } x \in \mathbb{R}^n - B(x^*,\rho_7). \end{cases}$$

For $\mu > 0$, $\Xi(x) = O(\|f(x)\|^{\mu})$, since $\bar{\mu} \geq \mu$. Thus, by reasoning similar to that used in the proof of Theorem (2.2.34), the assertions of (21) follow from Theorem (2.1.62). ∎

Assume $\bar{\mu} = 0$ but that the other hypotheses of Theorem (21) remain valid. With only a few changes in the above proof we could show that if $\delta_1, \delta_2 > 0$ are sufficiently small, then for any $\gamma > 0$ there exists $\rho > 0$ such that if Algorithm (1) runs with $x_0 \in B(x^*,\rho)$, then the sequence $x_0, x_1, \ldots$ of major iterates produced converges linearly to x^*. In this case we again appeal to Theorem (2.1.62), but with a $\Xi: \mathbb{R}^n \to [0,\infty]$ which does not vanish at x^*.

In practice, Brown's method is usually implemented on machines with finite, fixed-precision arithmetic. In case both single- and double-precision arithmetic are available and a nonsingular zero x^* of a "nice" f (whose derivative f' satisfies a Lipschitz condition of order 1) is to be approximated to nearly double-precision accuracy by Brown's method, then the above analysis suggests that it would suffice to do the housekeeping work of computing the $u_j^{(k)}$ in single-precision arithmetic while evaluating the difference quotients $\frac{\Delta f_k}{\Delta u}$ and computing the minor iterates $y_k = y_{k-1} - \frac{f_k(y_{k-1})}{\omega_k^{(k)}} u_k^{(k)}$ in double precision.

Rather than picking h as in step (1c), in the practical case when fixed-precision arithmetic is used it is probably best to choose h with an eye towards minimizing the sum of the roundoff and truncation errors. Assume, as suggested in §2.2, that $\omega_j^{(k)}$ is computed as $\|u_j^{(k)}\| \dfrac{\Delta f_k}{\Delta(u_j^{(k)}/\|u_j^{(k)}\|)}(y_{k-1},h)$ rather than $\dfrac{\Delta f_k}{\Delta u_j^{(k)}}(y_{k-1},h)$. If f satisfies a Lipschitz condition of order 1, then the absolute truncation error is then bounded by κh. On the other hand, if E bounds the absolute error in $fl(f_k(y_{k-1}+hu_j^{(k)}/\|u_j^{(k)}\|) - f_k(y_{k-1}))$, then $\frac{E}{h}$ bounds the absolute roundoff error in $\omega_j^{(k)}$. Let $\phi(h) = \kappa h + \frac{E}{h}$: then ϕ is minimized over $(0,\infty)$ at $h = \sqrt{E/\kappa}$. If estimates $\tilde{E}$ for E and $\tilde{\kappa}$ for κ are available, then $\sqrt{\tilde{E}/\tilde{\kappa}}$ is probably a good choice for h. A similar strategy was used in the numerical tests discussed in §3.6 of the minimization algorithms proposed in §3; the latter strategy (which depends on the cube root of $\tilde{\kappa}$ and so is less sensitive to errors in guessing $\tilde{\kappa}$) seemed to give satisfactory results.

2.4 Brent's Variation of Brown's Method

We deal again in this subsection with a version of Algorithm (2.1.54) which chooses $Q = I$, whence $f \equiv Q^T\underline{f} = \underline{f}$.

In Brown's method, the basis $u_k^{(k)},\ldots,u_n^{(k)}$ chosen for $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}^\perp$ can be very poorly conditioned. Brent (1973) has proposed that an orthonormal basis $u_k^{(k)},\ldots,u_n^{(k)}$ for $\mathrm{Span}\{w_1,\ldots,w_{k-1}\}^\perp$ be chosen. With such a choice, $U_k = [u_k^{(k)},\ldots,u_n^{(k)}]$ satisfies $U_k^T U_k = I$ and hence $\chi(U_k) = 1$. Moreover, the w_k produced in step (2.1.54k) of Algorithm (2.1.54) is simply

$$w_k = U_k(U_k^T U_k)^{-1}\begin{pmatrix}\omega_k^{(k)}\\ \vdots \\ \omega_n^{(k)}\end{pmatrix} = \sum_{j=k}^{n} \omega_j^{(k)} u_j^{(k)}.$$

Brent proposed choosing $p_k = w_k$. This choice has great intuitive appeal, for if H_j denotes the hyperplane in which $F_j(x) = f_j(y_{j-1}) + w_j^T(x-y_{j-1})$ vanishes, then $p_k = w_k$ causes y_k to be selected as the point in $\bigcap_{j=1}^{k} H_j$ closest to y_{k-1} in the 2-norm, so that $\|y_k-y_{k-1}\|$ is as small as possible. In nonlinear problems, linear approximations are only good locally; it therefore behooves us to attempt keeping $\|y_k-y_{k-1}\|$ as small as possible in order that the previously computed linearizations remain valid as long as possible.

Putting these ideas together, we obtain

(1) Algorithm: [Brent's method T_1 of (Brent, 1973)];

(1a) Let $f \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $\gamma > 0$ be given;

OUTERLOOP:

DO i = 1 to ∞;

(1b) If $f(x_{i-1}) = 0$ then stop and let $x_j = x_{i-1}$ for all $j \geq i$;

(1c) Choose $h \in (0,\gamma\|f(x_{i-1})\|]$;

(1d) $y_0 \leftarrow x_{i-1}$;

(1e) Let $u_1^{(1)},\ldots,u_n^{(1)}$ be an orthonormal basis for $\mathbb{R}^n$;

INNERLOOP:

DO k = 1 TO n;

(1f) $\omega_j^{(k)} \leftarrow \dfrac{\Delta f_k}{\Delta u_j^{(k)}}(y_{k-1},h)$ for $k \leq j \leq n$;

(1g) $p_k \leftarrow \sum_{j=k}^{n} \omega_j^{(k)} u_j^{(k)}$;

(1h) $\pi_k \leftarrow \|p_k\|^2 = \sum_{j=k}^{n} (\omega_j^{(k)})^2$;

(1i) $y_k \leftarrow y_{k-1} - \left(\dfrac{f_k(y_{k-1})}{\pi_k}\right) p_k$;

(1j) Choose orthonormal $u_k^{(k+1)},\ldots,u_n^{(k+1)}$ such that

Span $\{u_k^{(k+1)},u_{k+1}^{(k+1)},\ldots,u_n^{(k+1)}\} = \text{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\}$ and

p_k is a multiple of $u_k^{(k+1)}$;

END INNERLOOP;

(1k) $x_i \leftarrow y_n$;

END OUTERLOOP and Algorithm (1);

In step (1e) it suffices to choose $u_j^{(1)} = e_j$. Brent suggested that this choice be made for $i = 1$, i.e. during the first major iteration, while in subsequent major iterations $u_j^{(1)}$ equal the $u_j^{(j+1)}$ from the previous major iteration. However, under finite-precision arithmetic this latter choice invites accumulated roundoff error, and even with exact a arithmetic I see no clear advantage in this choice.

Step (1j) may be carried out as follows: find an orthogonal matrix $\Omega \in \mathcal{L}(\mathbb{R}^{n-k+1})$ such that $\Omega \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix} = \pm \|p_k\| e_1$ and choose

$[u_k^{(k+1)},\ldots,u_n^{(k+1)}] = U_k\Omega^T$, where $U_k = [u_k^{(k)},\ldots,u_n^{(k)}]$. Since

$$p_k = U_k \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix} = U_k\Omega^T\Omega \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix} = \pm \|p_k\|(U_k\Omega^T)e_1$$ and the columns

of $U_k\Omega$ are orthonormal and are linear combinations of $u_k^{(k)},\ldots,u_n^{(k)}$, this choice meets the conditions of step (1j).

The calculations just described for step (1j) may be conveniently carried out by means of a single Householder transformation or a product of Givens transformations. If this is done and the $u_j^{(k)}$ are maintained explicitly, then $n^2 + \mathcal{O}(n)$ storage locations will be required. It would be possible to reduce the storage to $n^2/2 + \Theta(n)$ by starting with $u_j^{(1)} = e_j$ and maintaining $[u_1^{(2)},\ldots,u_{k-1}^{(k)},u_k^{(k)},\ldots,u_n^{(k)}]$ implicitly as a product of Householder transformations. However, this scheme would increase from $\Theta(n^3)$ to $\Theta(n^4)$ the number of housekeeping arithmetic operations per major iteration needed to produce the $u_j^{(k)}$.

Powell (1968) has proposed a stable scheme which allows us to explicitly maintain the $u_j^{(k)}$ using only $n^2/2 + O(n)$ storage locations and $O(n^3)$ arithmetic operations per major iteration. Let $v_j = u_j^{(k)}$ and $z_j = \omega_j^{(k)}$, $k \le j \le n$. Given orthonormal vectors $v_k,\ldots,v_n$ and scalers $z_k,\ldots,z_n$, not all of which vanish, we thus wish to find orthonormal vectors $u_j^{(k+1)} = \hat{v}_j$, $k \le j \le n$, such that $\hat{v}_k$ is a multiple of $\sum_{j=k}^{n} z_j v_j$. Powell suggested doing this by

(2) <u>Algorithm</u>: [Powell's orthogonalization method];

(2a) Let orthonormal $v_k,\ldots,v_n \in \mathbb{R}^n$ and scalers $z_k,\ldots,z_n \in \mathbb{R}$ be given (with $z_j \neq 0$ for at least one j);

(2b) $\quad \nu \leftarrow \max\{j \,|\, z_j \neq 0, \quad k \leq j \leq n\};$

(2c) $\quad \hat{v}_j \leftarrow v_j \quad$ for $\quad n \geq j > \nu;$

$$\text{(2d)} \quad \hat{v}_j \leftarrow \frac{\left(\sum_{i=j}^{n} z_i^2\right) v_{j-1} - z_{j-1}\left(\sum_{i=j}^{n} z_i v_i\right)}{\sqrt{\left(\sum_{i=j}^{n} z_i^2\right)\left(\sum_{i=j-1}^{n} z_i^2\right)}} \quad \text{for} \quad \nu \geq j > k;$$

$$\text{(2e)} \quad \hat{v}_k \leftarrow \frac{\sum_{i=k}^{n} z_i v_i}{\sqrt{\sum_{i=k}^{n} z_i^2}} \quad ;$$

END Algorithm (2);

Powell noted that steps (2d) and (2e) **may** be carried out **as fol**lows (with $\sigma_j \in \mathbb{R}^n$ and $s_j \in \mathbb{R}$):

(3a) $\quad s_\nu \leftarrow z_\nu^2 ; \quad \sigma_\nu \leftarrow z_\nu v_\nu ;$

DO j = ν TO k+1 BY -1;

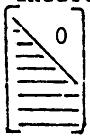
(3b) $\quad s_{j-1} \leftarrow s_j + z_{j-1}^2;$

$$\text{(3c)} \quad \hat{v}_j \leftarrow \frac{s_j v_{j-1} - z_{j-1}\sigma_j}{\sqrt{s_j s_{j-1}}} \; ;$$

(3d) $\quad \sigma_{j-1} \leftarrow \sigma_j + z_{j-1} v_{j-1};$

END;

(3e) $\quad v_k \leftarrow \sigma_k / \sqrt{s_k} \;\;;$

If the $u_j^{(k)}$ are updated by Powell's orthogonalization method starting with $u_j^{(1)} = e_j$, then we see inductively that $e_\ell^T u_j^{(k)} = 0$ for $1 \leq \ell \leq j-k$, whence U_k has the form . Steps (3c) and (3d) may be carried out in parallel in such a way that $u_j^{(k)}$ and $u_{j+1}^{(k+1)}$ occupy the same storage. Thus only $n^2/2 + \mathcal{O}(n)$ storage locations are required. Moreover, $n^3 + \mathcal{O}(n^2)$ arithmetic operations are required per major iteration, about the same number as when Householder transformations are used. If Givens transformations are used, then maintaining the $u_j^{(k)}$ costs $2n^3 + \mathcal{O}(n^2)$ arithmetic operations. (In these operation counts, one arithmetic operation consists of one multiplication and one addition.)

Powell explains in (Powell, 1968) why his orthogonalization method is stable.

Let us see how Algorithm (1) behaves when $f(x) = Ax - b$ is linear with $A \in \mathcal{L}(\mathbb{R}^n)$ nonsingular. Let

$$\mathcal{U}^{(k)} = [u_1^{(2)}, \ldots, u_{k-1}^{(k)}, u_k^{(k)}, \ldots, u_n^{(k)}]$$

with $\mathcal{U}^{(1)} = [u_1^{(1)}, \ldots, u_n^{(1)}]$. Steps (1g) and (1j) imply that for some orthogonal $\Omega_k \in \mathcal{L}(\mathbb{R}^{n-k+1})$ we have

$$\Omega_k \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix} = \pm\|p_k\| e_1 = \pm\|w_k\| e_1 \qquad \text{and}$$

$$\mathcal{U}^{(k+1)} = \mathcal{U}^{(k)} \begin{pmatrix} I_{k-1} & 0 \\ 0 & \Omega_k^T \end{pmatrix} .$$

Step (1f) and induction on k hence imply

$$L^{(k)} \equiv A\mathcal{U}^{(k)} = \begin{bmatrix} \pm\|w_1\| & 0 & 0 & \cdots & \cdots & 0 \\ X & \pm\|w_2\| & 0 & \ddots & & \vdots \\ X & X & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & X & \pm\|w_{k-1}\| & 0 & \cdots\ 0 \\ \vdots & \vdots & \vdots & X & \omega_k^{(k)} & \omega_n^{(k)} \\ \vdots & \vdots & \vdots & X & X & X \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X & X \cdots X & & X & X & \cdots\ X \end{bmatrix}$$

where X denotes a value that is not explicitly computed. Now the matrices $\mathcal{U}^{(k)}$ are all orthogonal, so $A = L^{(n)}\mathcal{U}^{(n)T}$ expresses A as the product of a lower triangular and an orthogonal matrix. In this respect, Algorithm (1) is a generalization of the "LQ" method for solving linear equations. Note that only the diagonal elements of $L = L^{(n)}$ are actually computed.

When step (2.2.12i) is used, Brown's method requires less overhead than the best implementation of Algorithm (1) described above: $n^3/6 + \mathcal{O}(n^2)$ arithmetic operations per major iteration, as opposed to $n^3 + \mathcal{O}(n^2)$ for (1), and $n^2/4 + \mathcal{O}(n)$ storage locations, as against $n^2/2 + \mathcal{O}(n)$ for (1). However, because its bases $u_k^{(k)}, \ldots, u_n^{(k)}$ are better conditioned and because its search directions make smaller angles with w_k (in the sense that $|p_k^T w_k|/(\|p_k\|\|w_k\|) = 1$ is larger), Algorithm (1) should enjoy a larger region of local convergence. Regarding its convergence, we may state

(4) Theorem: Theorem (2.2.34) holds for Brent's Algorithm (1).

Proof: As noted above, $w_k = p_k$ in Algorithm (1), where w_k is defined in (2.1.54k). By induction on step (1j) we have $\mathrm{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\} = \mathrm{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}$; since p_k is a linear combination of $u_k^{(k)},\ldots,u_n^{(k)}$, (2.1.54f(i)) holds with $\delta = 0$.

Because $\chi(U_k) = \|U_k\| = 1$, (2.1.54i) is satisfied with $c_0 = 1$.

Since $\|u_j^{(k)}\| = 1$, (1.2.13) implies

(5) $$\left|\omega_j^{(k)} - \frac{\partial f_k}{\partial u_j^{(k)}}(y_{k-1})\right| \le \kappa h^{\mu}$$

whenever y_0 and y_{k-1} lie close enough to x^*, in which case (2.1.54j) holds with $\epsilon_4 \xi = \kappa h^{\mu}$.

$$\text{Now } \frac{\partial f_k}{\partial p_k}(y_{k-1}) = f_k'(y_{k-1})p_k = \sum_{j=k}^{n} \omega_j^{(k)} f_k'(y_{k-1})u_j^{(k)} = \sum_{j=k}^{n} \omega_j^{(k)} \frac{\partial f_k}{\partial u_j^{(k)}}.$$

From (5) and step (1h) we thus have

$$\left|\frac{\partial f_k}{\partial p_k}(y_{k-1}) - \pi_k\right| = \left|\sum_{j=k}^{n} \omega_j^{(k)}\left(\frac{\partial f_k}{\partial u_j^{(k)}} - \omega_j^{(k)}\right)\right|$$

$$\le \kappa h^{\mu} \sum_{j=k}^{n} |\omega_j^{(k)}|$$

$$\le \sqrt{n}\,\kappa h^{\mu}\|p_k\|,$$

whence (2.1.54g) holds with $\epsilon_1 \xi = \sqrt{n}\,\kappa h^{\mu}$ whenever $\|y_0 - x^*\|$ and $\|y_{k-1} - x^*\|$ are sufficiently small.

Since $p_k = w_k$, (2.2.18) holds with $\eta_1 = 1$ and $\rho_3 = +\infty$.

The rest of the proof is like that of Theorem (2.2.34). ∎

2.5 Craig's Method

In this subsection we restrict our attention to linear functions $\underline{f}(x) = Ax - b$ and derive a version of Algorithm (2.1.3) in which q_k is chosen dynamically. The method derived, Craig's method, only requires storage for four vectors and the ability to compute Ax and A^Tx; it might turn out to be a good way to solve certain sparse linear systems.

When a version of Algorithm (2.1.54) is used to find a zero of a general nonlinear function $\underline{f} \in C^1(\mathbb{R}^n)$, the most reasonable choice for q_k in step (2.1.54e) seems to be a standard unit vector e_j, since extra function evaluations would be necessary to evaluate $f_k \equiv q_k^T\underline{f}$ if q_k had more than one nonzero component. When $\underline{f} = Ax - b$ is linear, however, this objection loses importance and we may consider other choices for q_k. If exact calculations (i.e. $\xi = 0$) are performed in the linear case, then Algorithm (2.1.54) has $x_j = x^*$ for all $j \geq 1$, so it suffices to consider Algorithm (2.1.3).

Assume for the rest of this subsection that

(1) $$\underline{f}(x) = Ax - b$$

where $A \in \mathcal{L}(\mathbb{R}^n)$ is nonsingular. For convenience, let

(2) $$r_k = \underline{f}(y_{k-1}) = Ay_{k-1} - b$$

denote the k^{th} residual. Notice that $f_k \equiv q_k^T\underline{f}$ has

$\frac{\partial f_k}{\partial p_k}(y) = f'(y)p_k = q_k^T A p_k$ and $\nabla f_k(y) = (f_k'(y))^T = A^T q_k$, independent of y. With exact arithmetic, i.e. $\xi = 0$, Algorithm (2.1.3) reduces in this case to

(3) Algorithm: [Algorithm (2.1.3) for $\underline{f}(x) = Ax - b$ with exact arithmetic];

(3a) Let nonsingular $A \in \mathcal{L}(\mathbb{R}^n)$, $b, y_0 \in \mathbb{R}^n$ be given;

LOOP:

DO $k = 1$ TO n;

(3b) Choose $q_k \in \mathrm{Span}\{q_1, \ldots, q_{k-1}\}^\perp$ with $\|q_k\| = 1$ and let

$f_k \equiv q_k^T \underline{f}$, i.e. $f_k(x) = q_k^T(Ax-b)$;

Let $W_{k-1} = [w_1, \ldots, w_{k-1}]$ and $P_{k-1} = I - W_{k-1}W_{k-1}^+$;

(3c) Choose $p_k \neq 0$ with

(3c(i)) $p_k \in \mathrm{Span}\{w_1, \ldots, w_{k-1}\}^\perp$, i.e. $P_{k-1}p_k = p_k$, and

(3c(ii)) $\frac{\partial f_k}{\partial p_k} = q_k^T A p_k \neq 0$;

(3d) $\pi_k \leftarrow q_k^T A p_k$;

(3e) $y_k \leftarrow y_{k-1} - \frac{f_k(y_{k-1})}{\pi_k} p_k = y_{k-1} - \frac{q_k^T r_k}{\pi_k} p_k$;

(3f) $w_k \leftarrow P_{k-1} A^T q_k$;

END LOOP and Algorithm (3);

Notice from (1), (2), and (3b,d,e) that

$$\underline{f}(y_k) = r_{k+1} = r_k - \frac{f_k(y_{k-1})}{\pi_k}Ap_k = r_k - \frac{q_k^T r_k}{q_k^T A p_k}Ap_k, \tag{4}$$

whence $f_k(y_k) = q_k^T r_{k+1} = 0$. Since

$$\mathrm{Span}\{w_1,\ldots,w_{k-1}\} = \mathrm{Span}\{A^T q_1,\ldots,A^T q_{k-1}\},$$

(3c(i)) implies $q_j^T A p_k = 0$, $1 \le j \le k-1$, so by (3b) and (4), $f_j(y_k) = f_j(y_{k-1})$ for $1 \le j \le k-1$. By induction we thus have

$$f_j(y_k) = q_j^T r_{k+1} = 0, \qquad 1 \le j \le k \le n. \tag{5}$$

Probably the most reasonable alternative to choosing $q_k = e_k$ is to choose

$$q_k = r_k / \|r_k\|, \tag{6}$$

provided $r_k \ne 0$. (If $r_k = 0$ then $y_{k-1} = x^* = A^{-1}b$ and we may stop.) From (5) we see that (6) does indeed result in a set of orthonormal vectors $q_1,\ldots,q_k$. Assume $r_k \ne 0$, $1 \le k \le n$, so that $q_1,\ldots,q_n$ are generated. Choice (6) implies $f_j(y_{k-1}) = q_j^T r_k = 0$ for $j \ne k$, so for $Q = [q_1,\ldots,q_n]$, $f = Q^T \underline{f}$ satisfies $f(y_{k-1}) = \|r_k\| e_k$, $1 \le k \le n$. But (3e) implies $\underline{f}(y_k) = r_{k+1} = r_k - \left(\frac{q_k^T r_k}{\pi_k}\right) Ap_k$, whence $f(y_k) - f(y_{k-1}) = \|r_{k+1}\| e_{k+1} - \|r_k\| e_k$ is a multiple of $Q^T A p_k$ (with $e_{n+1} = 0$). Thus when Algorithm (3) runs with q_k chosen by (6) and any choice of p_k satisfying (3c), $Q^T A[p_1,\ldots,p_n]$ is lower bidiagonal, i.e.

$$(7)\qquad Q^TA[p_1,\ldots,p_n] = \begin{bmatrix} \alpha_1 & 0 & \cdots & \cdots & 0 \\ \beta_2 & \alpha_2 & \ddots & & \vdots \\ 0 & \beta_3 & \alpha_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}$$

for certain α_i, β_i.

From (3c(i)) and (3f) we have $q_k^TAp_k = w_k^Tp_k$. Since $w_k \in \mathrm{Span}\{w_1,\ldots,w_{k-1}\}^\perp$ by (3f), the surest way to make (3c(ii)) hold is to choose p_k to be a multiple of w_k. With this choice, $Q^TA[w_1,\ldots,w_n]$ has the same form as RHS(7); hence

$$w_j^TA^Tq_k = q_k^TAw_j = 0 \qquad \text{for } j < k-1,$$

so

$$w_k \equiv P_{k-1}A^Tq_k = (I - [w_{k-1}][w_{k-1}]^+)A^Tq_k$$

$$= A^Tq_k - \left(\frac{w_{k-1}^TA^Tq_k}{\|w_{k-1}\|^2}\right) w_{k-1}$$

for $k > 1$, while $w_1 = A^Tq_1$. Since p_{k-1} is a multiple of w_{k-1} we thus have

$$(8)\qquad w_k = A^Tq_k - \left(\frac{q_k^TAp_{k-1}}{\|p_{k-1}\|^2}\right) p_{k-1}, \qquad 2 \le k \le n.$$

Since $q_k = r_k/\|r_k\|$, the most convenient choice for p_k is $p_k = \|r_k\|w_k$. By (6) and (8) we then have

(9a) $$p_1 = A^T r_1 \quad \text{and}$$

(9b) $$p_k = A^T r_k - \left(\frac{r_k^T A p_{k-1}}{\|p_{k-1}\|^2}\right) p_{k-1}, \qquad 2 \le k \le n.$$

With choices (6) and (9) for q_k and p_k, in (3d) and (3e) we have $\dfrac{f_k(y_{k-1})}{\pi_k} = \dfrac{q_k^T r_k}{q_k^T A p_k} = \dfrac{\|r_k\|^2}{r_k^T A p_k} = \dfrac{\|r_k\|^2}{\|p_k\|^2}$, since $p_k^T p_{k-1} = 0$ and $\|p_k\|^2 = p_k^T A^T r_k$ by (9). Thus, with these choices, Algorithm (3) becomes

(10) <u>Algorithm</u>: [Craig's method for solving $Ax^* = b$, version 1];

(10a) Let $b, y_0 \in \mathbb{R}^n$ and nonsingular $A \in \mathcal{L}(\mathbb{R}^n)$ be given;

LOOP:

DO $k = 1$ TO n;

(10b) $r_k \leftarrow Ay_{k-1} - b$;

(10c) If $r_k = 0$ then stop $[x^* = y_{k-1}]$;

(10d) $\tilde{p}_k \leftarrow A^T r_k$;

(10e) If $k = 1$ then $p_1 \leftarrow \tilde{p}_k$;

Else $p_k \leftarrow \tilde{p}_k - \left(\dfrac{\tilde{p}_k^T p_{k-1}}{\|p_{k-1}\|^2}\right) p_{k-1}$;

(10f) $$y_k \leftarrow y_{k-1} - \frac{\|r_k\|^2}{\|p_k\|^2} p_k;$$

END LOOP and Algorithm (10) [with $x^* = y_n$];

Algorithm (10) turns out to be a version of Craig's method (Craig, 1955) which differs only slightly from the version described by Faddeev and Faddeeva (1963, p. 405, display (23)). Indeed, as seen by comparing (10) with Faddeev and Faddeeva's version, step (10e) can be simplified. Assume $2 \le k \le n$. As above, we see from (9) that $r_{k-1}^T A p_{k-1} = \|p_{k-1}\|^2$.

Now $r_k = r_{k-1} - \dfrac{\|r_{k-1}\|^2}{\|p_{k-1}\|^2} A p_{k-1}$, so

$$\frac{\|r_k\|^2}{\|r_{k-1}\|^2} = 1 - \frac{2r_{k-1}^T A p_{k-1}}{\|p_{k-1}\|^2} + \frac{\|r_{k-1}\|^2 \|Ap_{k-1}\|^2}{\|p_{k-1}\|^4} = \frac{\|r_{k-1}\|^2 \|Ap_{k-1}\|^2}{\|p_{k-1}\|^4} - 1 \quad \text{and}$$

$$\frac{r_k^T A p_{k-1}}{\|p_{k-1}\|^2} = \frac{r_{k-1} A p_{k-1}}{\|p_{k-1}\|^2} - \frac{\|r_{k-1}\|^2 \|Ap_{k-1}\|^2}{\|p_{k-1}\|^4} = 1 - \frac{\|r_{k-1}\|^2 \|Ap_{k-1}\|^2}{\|p_{k-1}\|^4} .$$

Thus in step (10e) we have

$$\frac{\tilde{p}_k^T p_{k-1}}{\|p_{k-1}\|^2} = \frac{r_k^T A p_{k-1}}{\|p_{k-1}\|^2} = - \frac{\|r_k\|^2}{\|r_{k-1}\|^2} .$$

Incorporating this into Algorithm (10) and adding a stopping tolerance σ, we obtain what might be a good way to solve certain large, sparse systems of linear equations:

(11) <u>Algorithm</u>: [Craig's method for (approximately) solving $Ax^* = b$, version 2];

(11a) Let $b, y_0 \in \mathbb{R}^n$, $\sigma \in [0,\infty)$, and nonsingular $A \in \mathcal{L}(\mathbb{R}^n)$ be given;

(11b) Let $p_0 = 0$, $\|r_0\| = 1$;

LOOP:

DO $k = 1$ TO n;

(11c) $r_k \leftarrow Ay_{k-1} - b$;

(11d) If $\|r_k\| \leq \sigma$ then stop $[x^* \approx y_{k-1}]$;

(11e) $p_k \leftarrow A^T r_k + (\|r_k\|^2/\|r_{k-1}\|^2)p_{k-1}$;

(11f) $y_k \leftarrow y_{k-1} - (\|r_k\|^2/\|p_k\|^2)p_k$;

END LOOP and Algorithm (11) $[\text{with } x^* = y_n]$;

Algorithm (11) requires only the ability to compute Ay_{k-1} and $A^T r_k$ as well as storage for four vectors: b, y_k, p_k, and r_k or $A^T r_k$. We could eliminate the storage for b (as Faddeev and Faddeeva do) by changing (11c) to: $r_k \leftarrow r_{k-1} - (\|r_{k-1}\|^2/\|p_{k-1}\|^2)Ap_{k-1}$ (for $k \geq 2$); but this would invite a growing numerical error in the value of r_k under finite precision arithmetic.

If an approximate solution suffices, then a choice of $\sigma > 0$ might result in fewer than n iterations. (Even if $\sigma = 0$, it is possible for Algorithm (11) to stop after fewer than n iterations, but this is exceedingly unlikely.) On the other hand, if inexact arithmetic is used and a very accurate solution is required, then it might be necessary to improve y_n by setting $y_0 \leftarrow y_n$ and repeating Algorithm (11). We might also consider allowing the LOOP to continue executing with values of $k > n$. If this is done, however, convergence might actually be slowed. Algorithm (11) is intimately related to the (linear) conjugate gradient algorithm; in fact, Faddeev and Faddeeva derive Craig's method by applying the method of conjugage gradients to the problem $AA^T z^* = b$ (whose solution z^* is related to that of $Ax^* = b$ by the change of vsriables $x = A^T z$). Fletcher and Reeves (1964) have observed with their nonlinear version of the conjugate gradient method that periodic restarting leads to much better performance; this amounts to setting $y_0 \leftarrow y_k$ (with $k \geq n$) and starting over again in Algorithm (11). Fletcher and Reeves state that when the linear conjugate gradient method is carried out using inexact arithmetic, an $(n+1)^{\underline{st}}$ iteration helps reduce the accumulated roundoff error. Thus, when inexact arithmetic is used and an accurate solution is desired, it may be well to extend the limits of the LOOP in Algorithm (11) to "1 TO n+1".

If A is dense, then Algorithm (11) requires $2n^3 + \mathcal{O}(n^2)$ arithmetic operations (1 operation = 1 add + 1 multiply) to solve $Ax^* = b$, whereas Gaussian elimination requires but $n^3/3 + \mathcal{O}(n^2)$. Craig's method might be advantageous, however, in some cases when A is large and sparse.

Assume momentarily that $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$, $b \in \mathbb{R}^m$ with $m > n$. The stopping test (11d) suggests that Algorithm (11) might be used to (approximately) solve the least squares problem: minimize $\|Ax - b\|$. Paige (1974) describes an algorithm similar to (11) for this problem. He also discusses the close relationship between these algorithms, the bidiagonalization method of Golub and Kahan (1965, pp. 210-211), and Lanczos's method for tridiagonalizing a symmetric matrix.

2.6 Modifications

In this subsection we discuss some modifications to Algorithm (2.1.54) aimed at increasing its range of applicability and improving its efficiency. First we consider taking damped steps, then we note how the overhead may be reduced when some components of $\underline{f}$ are linear, and finally we discuss reusing p_k and π_k.

Damped Steps

Whereas the undamped Newton's method takes steps of the form $x_i = x_{i-1} - \underline{f}'(x_{i-1})^{-1}\underline{f}(x_{i-1})$ in an effort to solve $\underline{f}(x) = 0$, the damped Newton's method selects a damping factor d_i during the i^{th} iteration, usually with $d_i \in (0,1]$, and takes the step $x_i = x_{i-1} - d_i\underline{f}'(x_{i-1})^{-1}\underline{f}(x_{i-1})$. The damping factor d_i is often chosen in an attempt to maximize the decrease in some criterion function, such as $\|\underline{f}\|^2$ (or $\emptyset$ if $\underline{f} = \nabla\emptyset$). When x_0 lies outside a region of local convergence for Newton's method, the damped Newton's method is frequently able to converge while the undamped method diverges. By analogy with Newton's method, it may sometimes be wise to take damped steps in Algorithm (2.1.54).

Several ways suggest themselves for taking damped steps in Algorithm (2.1.54). The simplest may be to select a damping factor d_i at the start of the i^{th} major iteration, perhaps based on performance in the previous major iterations, and change step (2.1.54h) to:

$$\text{"Choose } y_k \in \bar{B}\left(y_{k-1} - d_i \left[\frac{f_k(y_{k-1})}{\pi_k}\right] p_k,\ \epsilon_2 \xi \|\underline{f}(y_0)\|\right);\text{"}.$$

Another way would be to do the above with a small $d_i > 0$, then replace step (5.1.54ℓ) by "$x_i \leftarrow x_{i-1} + \lambda_i(y_n - y_0)$;", where λ_i is chosen to approximately minimize an appropriate criterion function. A third alternative would be to select a maximum step length λ_{max} and replace step (2.1.54) by:

$$\text{"Choose } y_k \in \bar{B}\left(y_{k-1} - \left[\frac{f_k(y_{k-1})}{\pi_k}\right] \min\left\{1, \frac{\lambda_{max}}{\left|\frac{f_k(y_{k-1})}{\pi_k}\right| \|p_k\|}\right\} p_k,\ \epsilon_2 \|\underline{f}(y_0)\|\right);\text{"}.$$

The bound λ_{max} could be fixed or chosen anew each major iteration.

Using Linearity

Assume the first n_1 components $\underline{f}_1,\ldots,\underline{f}_{n_1}$ of $\underline{f} = (\underline{f}_1,\ldots,\underline{f}_n)^T \in C^1(\mathbb{R}^n)$ are linear, with $1 \leq n_1 < n$. In this case we may choose $q_j = e_j$ for $1 \leq j \leq n_1$ and perform the first n_1 minor iterations once and for all at the beginning, thus determining a point $x_1 = y_{n_1}$ and vectors $w_1,\ldots,w_{n_1}$ such that $\underline{f}_j(x) = 0$ for $1 \leq j \leq n_1$ whenever $x - x_1 \in \text{Span}\{w_1,\ldots,w_{n_1}\}^{\perp}$. The range of the inner loop can then be changed to "$k = n_1+1$ TO n", while the range of the outer loop becomes "$i = 2$ TO ∞" and (2.1.54c) is replaced by "$y_{n_1} \leftarrow x_{i-1}$". The basis $u_{n_1+1}^{(n_1+1)},\ldots,u_n^{(n_1+1)}$ for $\text{Span}\{w_1,\ldots,w_{n_1}\}^{\perp}$ may be chosen once and for all as soon as $w_1,\ldots,w_{n_1}$ have been determined. Of course, with inexact arithmetic it would be necessary to periodically repeat the first n_1 minor iterations.

Reusing p_k and π_k to Increase Efficiency

Many iterative methods which compute (approximate) derivatives, such as those considered here, may be readily modified to reuse some or all of these (approximate) derivatives for several succeeding iterations. Brent's main theme in (Brent, 1973) was that under suitable conditions it is possible to choose the number of times that old derivative information is to be reused in such a way that a lower bound on the efficiency of the process is maximized.

Suppose algorithm A produces the sequence $x_1,x_2,\ldots$ starting from x_0. If $\mathcal{W}_i$ denotes the work required to produce x_i from x_{i-1} (using any previously computed information available, with $\mathcal{W}_i$ measured in appropriate units) and if $\lim_{i\to\infty} x_i = x^*$, then Brent defines the efficiency of this process to be

$$E = \liminf_{i \to \infty} \left(\frac{\log(-\log\|x_i - x^*\|)}{\sum_{j=1}^{i} \mathcal{W}_i} \right).$$

He notes when "lim inf" may be replaced by "lim" that the higher the efficiency of a process, the less work will be required to achieve a specified and sufficiently small upper bound on $\|x_i - x^*\|$.

Under appropriate conditions, the efficiency of particular versions of Algorithm (2.1.54) may be increased if they are allowed to reuse the p_k and π_k--Brent had this in mind for Brown's method and his variation thereof. While we will not consider here the important question of maximizing the efficiency of the thus modified Algorithm (2.1.54) (see (Brent, 1973) for more discussion related to this question), we will bound the convergence rate of this algorithm. (Of course, the rate of convergence by itself says nothing about the efficiency of the process.) Let $\mathbb{N}^+ = \{1,2,3,\ldots\}$ and consider

(1) <u>Algorithm</u>: [generalized Brown's method, modified to reuse p_k, π_k];

(1a) Let $\underline{f} \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$, $\Xi: \mathbb{R}^n \to [0,\infty]$, $J \in \mathbb{N}^+$, and the constants c_0, δ, ϵ_1, ϵ_2, ϵ_4, η be given;

OUTERLOOP:

DO $i = 0$ TO ∞ BY J;

(1b) If $\underline{f}(x_i) = 0$ THEN Stop and let $x_j = x_i$ for all $j > i$;

(1c) $y_0^{(1)} \leftarrow x_i$;

(1d-k) Perform step (2.1.54d) and the innerloop of Algorithm (2.1.54), with "y_j" replaced by "$y_j^{(1)}$", and retain ξ, $p_1,\ldots,p_n$, $\pi_1,\ldots,\pi_n$, and $f = Q^T\underline{f}$;

REUSE:

DO j = 2 TO J;

(1ℓ) $y_0^{(j)} \leftarrow y_n^{(j-1)}$;

INNER_REUSELOOP:

DO k = 1 TO n;

(1m) Choose $y_k^{(j)} \in \bar{B}\left(y_{k-1}^{(j)} - \left[\frac{f_k(y_{k-1}^{(j)})}{\pi_k}\right] p_k, \quad \epsilon_2\xi\|\underline{f}(y_0^{(1)})\|\right)$;

END INNER_REUSELOOP and REUSE;

(1n) $x_{i+j} \leftarrow y_n^{(j)}$ for $1 \le j \le J$;

END OUTERLOOP and Algorithm (1);

Note that if $J=1$ (and i is replaced by $i+1$), then the REUSE loop is not executed and Algorithm (1) reduces to Algorithm (2.1.54).

Assume $\underline{f}'$ satisfies a Lipschitz condition of order $\mu \in [0,1]$ near a nonsingular zero x^* of $\underline{f}$. Theorem (2) below states that if $\|x_0-x^*\|$ is small enough and Ξ behaves properly, then the subsequence $x_0,x_J,x_{2J},\ldots$ of the sequence $x_0,x_1,x_2,\ldots$ generated by Algorithm (1) converges to x^* with $\mathcal{O}_Q(x_0,x_J,x_{2J},\ldots) \ge 1 + J\mu$. A slightly more involved proof shows that in fact the sequence $x_0,x_1,\ldots$ converges to x^* with J-step Q-order not less than $1+J\mu$, i.e. for any $j \in \mathbb{N}$, the sequence $x_j,x_{j+J},x_{j+2J},\ldots$ converges to x^* with $\mathcal{O}_Q(x_j,x_{j+J},\ldots) \ge 1 + J\mu$. This implies, in turn, that the sequence $x_0,x_1,\ldots$ converges to x^* with R-order $\mathcal{O}_R(x_0,x_1,\ldots) \ge (1+J\mu)^{1/J}$.

(2) Theorem: If there exists $c_1 \in (0,\infty)$ such that the basis vectors $u_j^{(k)}$ chosen in step (1i) satisfy $\|u_j^{(k)}\| \le c_1$, then for each $J \in \mathbb{N}^+$ Theorem (2.1.62) holds for Algorithm (1) with x_i replaced by x_{Ji} and "$i+\mu$" replaced by "$1+J\mu$".

Proof sketch: From Lemma (2.1.20) there exist $\rho_1 \in (0,\rho_2)$ and $\xi_0, K \in (0,\infty)$ such that if $\xi \le \xi_0$, $y_0^{(j)} \in \bar{B}(x^*, \min\{\|x_i-x^*\|, \rho_1\})$, and $y_1^{(j)},\ldots,y_{k-1}^{(j)} \in B(x^*,\rho_1)$, then $\|y_{k-1}^{(j)} - y_0^{(j)}\| \le K\|x_i-x^*\|$. Hence, under these conditions, $\|f_k'(y_{k-1}^{(j)}) - f_k'(y_0^{(j)})\| \le \kappa(K\|x_i-x^*\|)^\mu$, so for $2 \le j \le J$ the inner loop of the REUSE loop satisfies the conditions of Algorithm (2.1.3) with $\epsilon_1\xi$ replaced by $\epsilon_1\xi + \kappa(K\|x_i-x^*\|)^\mu$, $\epsilon_3\xi$ replaced by $\epsilon_3\xi + c_0c_1\kappa(K\|x_i-x^*\|)^\mu\sqrt{n}$, and, for small enough $\|x_i-x^*\|$, η replaced by $\eta/2$. Hence for suitably small $\|x_i-x^*\|$ we may redefine ξ and find by Lemma (2.1.40) that

$$\|y_n^{(j)}-x^*\| = \mathcal{O}(\|x_i-x^*\|^{1+j\mu}), \quad 1 \le j \le J. \qquad \blacksquare$$

3. Applying the Generalized Brown's Method to Unconstrained Minimization Problems

In this section we consider using particular versions of Algorithm (2.1.54) to help find a local minimizer x^* of $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$. Specifically, we propose new descent methods in which the minor iterates themselves are descent steps.

It is well known that if $\phi \in C^1(\mathbb{R}^n, \mathbb{R})$ and x^* is a local minimizer of ϕ, i.e. there exists $\rho > 0$ such that $\phi(x) \geq \phi(x^*)$ for all $x \in B(x^*, \rho)$, then $\nabla\phi(x^*) = 0$. Although the converse is false, we will nonetheless choose $\underline{f} = \nabla\phi$ in Algorithm (2.1.54) and run the risk of converging to a point that is not a local minimizer.

Corresponding to our assumption in §2 that $\underline{f} \in C^1(\mathbb{R}^n)$ with $\underline{f}(x^*) = 0$ and $\underline{f}'(x^*)$ nonsingular, we will usually assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R}^1)$ with $\nabla\phi(x^*) = 0$ and $\nabla^2\phi(x^*) = \phi''(x^*)$ positive definite. For brevity, we will call a point x^* at which these conditions hold a <u>nonsingular minimizer</u> of ϕ. As is well known, these conditions assure that x^* is an isolated local minimizer, i.e. there exists $\rho > 0$ such that $\phi(x) > \phi(x^*)$ whenever $x \in B(x^*, \rho)$ and $x \neq x^*$. Under these conditions we could simply apply Brown's method or Brent's variation thereof with $f = \nabla\phi$ and be assured of rapid convergence to x^* whenever $\|x_0 - x^*\|$ is small enough. However, we desire to construct globally convergent minimization algorithms and thus do away with the "sufficiently small" requirement on $\|x_0 - x^*\|$. Moreover, we wish to avoid converging to an x^* with $\phi(x^*) > \phi(x_0)$. Hence we will restrict our attention to descent algorithms, i.e. algorithms which assure $\phi(x_i) \leq \phi(x_{i-1})$ whenever $\nabla\phi(x_{i-1}) \neq 0$ and which periodically have $\phi(x_{i-1}) - \phi(x_i)$ large enough to prevent convergence to a noncritical point. With such a strategy, convergence to an isolated local maximizer is impossible (provided $\nabla\phi(x_0) \neq 0$) and, though possible, convergence to a saddle point seems unlikely.

One simple way to incorporate Brown's method into a descent algorithm would be as follows: given x_{i-1}, compute a candidate $\hat{x}_i$ for x_i by executing one major iteration of Brown's method with $f = \nabla\phi$; if $\phi(\hat{x}_i) < \phi(x_{i-1})$, then let $x_i = \hat{x}_i$; otherwise take one steepest descent step of the Armijo (1966) type from x_{i-1} and call the result x_i; to prevent convergence to a noncritical point, periodically compute a second candidate $\tilde{x}_i$ for x_i by taking an Armijo type steepest descent step from x_{i-1}, and choose $x_i = \tilde{x}_i$ if $\phi(\tilde{x}_i) < \phi(\hat{x}_i)$, even if $\phi(\hat{x}_i) < \phi(x_{i-1})$. A better way might be to choose a damping factor $d_i \in (0,1]$ at x_{i-1}, then compute $\bar{x}_i$ by executing one major iteration of Brown's method with $f = \nabla\phi$ and step (2.2.12h) replaced by $y_k \leftarrow y_{k-1} - d_k\left[f_k(y_{k-1})/\omega_k^{(k)}\right]u_k^{(k)}$, as suggested in §2.6, and proceed as before with either $\hat{x}_i = \bar{x}_i$ or $\hat{x}_i$ chosen by approximately minimizing ϕ in the direction $\bar{x}_i - x_{i-1}$ from x_{i-1}; the damping factor d_i should probably be determined dynamically from the past performance of the algorithm in such a way that $d_i \rightarrow 1$ near a nonsingular minimizer x^* of ϕ. The scheme using the damping factors is very similar to the damped Newton's method, in which x_i is chosen to approximately minimize ϕ in the direction $(\phi''(x_{i-1}))^{-1}\nabla\phi(x_{i-1})$ from x_{i-1}. In fact, because both take similar steps and both require about the same number of function evaluations per iteration (since ϕ'' is symmetric, and assuming that Brown's method is implemented properly), there may be little advantage in one over the other.

Assume until otherwise stated that $\phi \in C^2(\mathbb{R}^n, \mathbb{R}^1)$, $\nabla\phi(x^*) = 0$, and $\phi''(x^*)$ is positive definite.

The schemes just described for incorporating Brown's method into a descent algorithm have the disadvantage that a complete major iteration of Brown's method (undamped, or with a poorly chosen damping factor) might be carried out in vain when $\|x_{i-1}-x^*\|$ is large; thus much effort might be wasted. Even when $\|x_{i-1}-x^*\|$ is very small, there is no reason for the minor iterates $y_0,\ldots,y_n$ generated by Brown's method from x_{i-1} with $f = \nabla\phi$ to satisfy $\phi(y_k) \leq \phi(y_{k-1})$; the only thing of which we can be sure is that $\phi(y_n) < \phi(y_0)$ whenever $y_0 = x_{i-1}$ lies close enough to x^* (and the damping factor, if used, is properly chosen). We shall therefore restrict our attention to versions of Algorithm (2.1.54), undamped for the present, which always have $\phi(y_k) \leq \phi(y_{k-1})$ whenever $\xi = 0$ and $\|y_0-x^*\|$ is sufficiently small.

The versions of Algorithm (2.1.54) which we propose for use in minimization routines allow themselves to be combined quite naturally with slow but robust and globally convergent descent methods such as steepest descent. The resulting hybrid algorithm works as follows: at major iterate x_{i-1} the robust descent method chooses a search direction p_i; the version of (2.1.54) being used computes y_1 from $y_0 = x_{i-1}$ by performing part of the first minor iteration; if $\phi(y_0)-\phi(y_1)$ is too small, then the robust descent method determines x_i of the form $x_i = x_{i-1} - \lambda_i p_i$; but if $\phi(y_0) - \phi(y_1)$ is sufficiently large, then the first minor iteration is completed and enough of the second is carried out to determine y_2; in general, if $\phi(y_{k-1})-\phi(y_k)$ is sufficiently large, then the rest of the $k^{\underline{th}}$ minor iteration is completed and if $k < n$ then enough of the $(k+1)^{\underline{st}}$ is carried out to determine y_{k+1}, while if $k = n$ then $x_i = y_n$; but if $\phi(y_{k-1})-\phi(y_k)$ is too small and $k > 1$, then the choice $x_i = y_{k-1}$ is satisfactory, since $\phi(y_0)-\phi(y_{k-1}) \geq \phi(y_0)-\phi(y_1)$ was large enough to prevent convergence to a noncritical point. Whenever $\|x_{i-1}-x^*\|$ is small enough, all minor iterations will be successful and the hybrid algorithm will reduce to a version of Algorithm (2.1.54). Thus there is a natural transition from the slow but robust descent method to the rapidly but only locally convergent version of (2.1.54).

While the minimization algorithms proposed here assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R}^1)$, they are designed for use in cases where ϕ' and ϕ'' are not explicitly available. Indeed, since these algorithms work with $f = Q^T\nabla\phi$, where $Q = [q_1,\ldots,q_n]$ is generated anew each major iteration, it will often be more efficient to approximate $f_k = q_k^T\nabla\phi$ and $\frac{\partial f_k}{\partial u} = q_k^T\phi''u$ by finite differences than to employ explicitly computed values of $\nabla\phi$ and ϕ'' to compute these values.

3.1 Choosing q_k and p_k

In this subsection we consider how to choose q_k and p_k in Algorithm (2.1.54) in such a way that $\phi(y_k) < \phi(y_{k-1})$ near a local minimizer x^* of $\phi \in C^2(\mathbb{R}^n,\mathbb{R})$ at which $\phi''(x^*)$ is positive definite; as indicated in §3.0, we let $\underline{f} = \nabla\phi$ in Algorithm (2.1.54). Near such an x^*, ϕ behaves approximately as a quadratic function; hence it suffices for our purposes in this subsection to assume that

$$\phi(x) = \frac{1}{2}x^TAx - b^Tx \tag{1}$$

is quadratic with $A \in \mathcal{L}(\mathbb{R}^n)$ positive definite (and symmetric). In an effort to make a "sufficient" decrease in ϕ more likely for nonquadratic ϕ as well as to make possible a smooth and natural transition from a robust descent method to Algorithm (2.1.54), as described in §3.0, we shall insist in the present quadratic case that ϕ decrease along the entire line segment from y_{k-1} to y_k.

It suffices to consider the case $k = 1$, as we now proceed to show. Note that $\underline{f}' = \phi'' = A$; since $f_k = q_k^T f_k = \phi' q_k = \frac{\partial\phi}{\partial q_k}$, we have $\frac{\partial f_k}{\partial u} = \frac{\partial^2\phi}{\partial u \partial q_k} = q_k^T \phi'' u = q_k^T A u$ and $f_k' = \left(\frac{\partial f_k}{\partial e_1}, \ldots, \frac{\partial f_k}{\partial e_n}\right) = q_k^T A$, i.e. $\nabla f_k = Aq_k$. Defining $W_{k-1} = [w_1, \ldots, w_{k-1}]$ and $P_{k-1} = I - W_{k-1}W_{k-1}^+$ as in Algorithm (2.1.54), we assume in the present quadratic case that

$\omega_j^{(k)} = \frac{\partial f_k}{\partial u_j^{(k)}} = q_k^T A u_j^{(k)}$ is computed exactly, $k \leq j \leq n$, where $u_k^{(k)}, \ldots, u_n^{(k)}$ is a basis for $\mathrm{Span}\{w_1, \ldots, w_{k-1}\}^\perp$. Thus $U_k = [u_k^{(k)}, \ldots, u_n^{(k)}]$ satisfies $U_k U_k^+ = P_{k-1}$, so in step (2.1.54k) we have

$$
\begin{aligned}
w_k = U_k (U_k^T U_k)^{-1} \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix} &= U_k (U_k^T U_k)^{-1} U_k^T A q_k \\
&= U_k U_k^+ A q_k \\
&= P_{k-1} A q_k .
\end{aligned}
$$

By induction on k we thus find

(2) $$\mathrm{Span}\{w_1, \ldots, w_{k-1}\} = \mathrm{Span}\{Aq_1, \ldots, Aq_{k-1}\}.$$

(Note that $Aq_1, \ldots, Aq_{k-1}$ are linearly independent, $k \geq 2$, since $q_1, \ldots, q_{k-1}$ are orthonormal, hence linearly independent and A is positive definite, hence nonsingular. Thus $\mathrm{Span}\{Aq_1, \ldots, Aq_{k-1}\}^\perp$ has dimension $n-k+1$, as we tacitly claimed above.) Because of (2) we have

(3) $$\mathrm{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\} = \mathrm{Span}\{Aq_1,\ldots,Aq_{k-1}\}^{\perp},$$

which implies that $Aq_1,\ldots,Aq_{k-1},\ u_k^{(k)},\ldots,u_n^{(k)}$ span $\mathbb{R}^n$:

(4) <u>Lemma</u>: If $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ is positive definite, if $q_1,\ldots,q_{k-1} \in \mathbb{R}^n$ are linearly independent and likewise for $u_k^{(k)},\ldots,u_n^{(k)} \in \mathbb{R}^n$, and if (3) holds then

$$\mathrm{Span}\{q_1,\ldots,q_{k-1},u_k^{(k)},\ldots,u_n^{(k)}\} = \mathbb{R}^n.$$

<u>Proof</u>: It suffices to show that $M = [q_1,\ldots,q_{k-1},u_k^{(k)},\ldots,u_n^{(k)}]$ is nonsingular. Let $Q_{k-1} = [q_1,\ldots,q_{k-1}]$ and $U_k = [u_k^{(k)},\ldots,u_n^{(k)}]$. By (3) we have

$$M^TAM = \begin{bmatrix} Q_{k-1}^TAQ_{k-1} & 0 \\ 0 & U_k^TAU_k \end{bmatrix}.$$

Since the columns of Q_{k-1} and U_k are linearly independent and A is positive definite, both $Q_{k-1}^TAQ_{k-1}$ and $U_k^TAU_k$ are nonsingular, whence M^TAM and hence M are both nonsingular. ∎

Now in the k^{th} minor iteration of Algorithm (2.1.54) we are free to choose q_k and p_k arbitrarily except for the conditions

(5) $$q_k \in \mathrm{Span}\{q_1,\ldots,q_{k-1}\}^{\perp}, \quad \|q_k\| = 1,$$

(6) $$p_k \in \mathrm{Span}\{Aq_1,\ldots,Aq_{k-1}\}^{\perp}, \quad \left|\frac{\partial f_k}{\partial p_k}\right| = |q_k^TAp_k| \geq \eta_0\|p_k\| > 0$$

for some $\eta_0 > 0$: note from the proof of Lemma (2.1.15) that $\|P_{k-1}\nabla f_k\| = \|P_{k-1}Aq_k\| \geq 1/\|A^{-1}\|$, so we have taken $\eta_0 = \eta/\|A^{-1}\|$ with η from step (2.1.54f). Lemma (4) implies that any q_k satisfying (5) may be expressed in the form

$$q_k = \underline{q}_k + \bar{q}_k, \text{ where } \underline{q}_k \in \mathrm{Span}\{u_k^{(k)},\dots,u_n^{(k)}\} \text{ and } \bar{q}_k \in \mathrm{Span}\{q_1,\dots,q_{k-1}\}.$$

For any p_k satisfying (6) and any $\lambda \in \mathbb{R}$, $f_k(y_{k-1}+\lambda p_k)-f_k(y_{k-1})=\lambda q_k A p_k = \lambda \underline{q}_k A \underline{q}_k$.
Similarly, $\dfrac{\partial f_k}{\partial u_j^{(k)}} = q_k^T A u_j^{(k)} = \underline{q}_k^T A u_j^{(k)}$, $k \leq j \leq n$. Since for any $\underline{q}_k \in \mathrm{Span}\{u_k^{(k)},\dots,u_n^{(k)}\}$ there exists by Lemma (4) a $\bar{q}_k \in \mathrm{Span}\{q_1,\dots,q_{k-1}\}$ such that $\tilde{q}_k \equiv \underline{q}_k + \bar{q}_k \in \mathrm{Span}\{q_1,\dots,q_{k-1}\}^{\perp}$, it suffices to choose $\underline{q}_k \in \mathrm{Span}\{u_k^{(k)},\dots,u_n^{(k)}\} - \{0\}$, construct $\tilde{q}_k = (I - [q_1,\dots,q_{k-1}][q_1,\dots,q_{k-1}]^+)\underline{q}_k$, and let $q_k = \tilde{q}_k/\|\tilde{q}_k\|$, but work with $\|\tilde{q}_k\| f_k(x) = \underline{q}_k^T \nabla \phi(x) = \underline{q}_k^T(Ax-b)$. Note for $n=1$ that $\tilde{q}_1 = \underline{q}_1$, so $q_1 = \underline{q}_1/\|\underline{q}_1\|$.

So far as choosing p_k is concerned, let $\psi: \mathbb{R}^{n-k+1} \to \mathbb{R}$ be given by $\psi(z) = \phi(y_{k-1}+U_k z)$. By (6), p_k may be expressed as $p_k = U_k \hat{p}_k$, where $\hat{p}_k \in \mathbb{R}^{n-k+1}$. But for all $\lambda \in \mathbb{R}$, $\phi(y_{k-1}+\lambda p_k) = \psi(\lambda \hat{p}_k)$. Since (1) implies that

$$\psi(z) = \tfrac{1}{2} z^T (U_k^T A U_k) z - [(b - Ay_{k-1})^T U_{k-1}] z + \phi(y_{k-1})$$

is also quadratic, we see that the question of how to choose q_k and p_k

so that $\emptyset$ decreases along the line segment from y_{k-1} to y_k for $k > 1$ is equivalent to the same question for $k = 1$. Hence we now consider how to choose $\underline{q}_1$ and p_1.

We shall allow $\underline{q}_1$ to be chosen arbitrarily. Given $q_1 = \underline{q}_1/\|\underline{q}_1\|$, let

(7) $$r_1 = \nabla\emptyset(y_0) = Ay_0 - b.$$

Note that $f_1(y_0) = q_1^T\nabla\emptyset(y_0) = q_1^T r_1$ and $\pi_1 = \frac{\partial f_1}{\partial p_1}(y_0) = q_1^T A p_1$. Thus

(8) $$y_1 = y_0 - \left(\frac{f_1(y_0)}{\pi_1}\right)p_1 = y_0 - \left(\frac{q_1^T r_1}{q_1^T A p_1}\right)p_1.$$

Let

(9) $$r_2 = \nabla\emptyset(y_1) = Ay_1 - b = r_1 - \left(\frac{q_1^T r_1}{q_1^T A p_1}\right)Ap_1.$$

Since $\nabla\emptyset$ is linear with $\frac{\partial\emptyset}{\partial p_1}(y_0) = p_1^T r_1$ and $\frac{\partial\emptyset}{\partial p_1}(y_1) = p_1^T r_2$, $\emptyset$ decreases along the line segment from y_0 to y_1 iff

(10a) $$\left(\frac{q_1^T r_1}{q_1^T A p_1}\right)p_1^T r_1 \geq 0 \qquad \text{and}$$

(10b) $$\left(\frac{q_1^T r_1}{q_1^T A p_1}\right)p_1^T r_2 = \left(\frac{q_1^T r_1}{q_1^T A p_1}\right)p_1^T r_1 - \left(\frac{q_1^T r_1}{q_1^T A p_1}\right)^2 p_1^T A p_1 \geq 0.$$

Since A is positive definite, (10a) follows from (10b), so we need only assure that (10b) holds. Notice that (10b) holds with equality if p_1 is a multiple of q_1. For other choices of p_1, say as a linear combination of q_1, r_1, Aq_1, Ar_1, it would in general be necessary to compute $q_1^T Av$ for at least one vector v linearly independent of q_1 in order to assure that (10b) holds. In the nonquadratic case this would mean approximating $\frac{\partial^2 \phi}{\partial q_1 \partial v}$, which would cost extra function evaluations. Thus the most reasonable choice for p_1 appears to be $p_1 = q_1$, and this is the choice we shall make.

Recall the discussion above about choosing q_k. The minimization algorithms we propose choose p_k and q_k according to

(11a) Choose $p_k \in \mathrm{Span}\{u_k^{(k)}, \ldots, u_n^{(k)}\} - \{0\}$;

(11b) Let $Q_{k-1} = [q_1, \ldots, q_{k-1}]$,

(11c) $\tilde{q}_k = (I - Q_{k-1}Q_{k-1}^+)p_k$, and

(11d) $q_k = \tilde{q}_k / \|\tilde{q}_k\|$.

(For $k = 1$, $I - Q_0 Q_0^+ = I$.) Note that $\tilde{q}_k \neq 0$ by Lemma (4). Since $Q_{k-1}^T Ap_k = 0$, this choice results in $\left|\frac{\partial f_k}{\partial p_k}\right| = |q_k^T Ap_k| = p_k^T Ap_k / \|\tilde{q}_k\|$; since $\|p_k\| \geq \|\tilde{q}_k\|$, we have $\left|\frac{\partial f_k}{\partial p_k}\right| \geq \|p_k\| \sigma_{min}$, where $\sigma_{min} = \|A\| / \chi(A)$ is the least singular value of A. Thus (5) and (6) are both satisfied.

In §3.5 we will discuss some specific ways to generate $u_k^{(k)}, \ldots, u_n^{(k)}$ and choose p_k in (11a).

3.2 The Local Form of the Proposed Minimization Algorithms

In this subsection we state the algorithm to which the proposed minimization algorithms reduce near a nonsingular minimizer x^* and bound the convergence rate of this algorithm. We also discuss how some of the required calculations may be carried out. Since we expect some of these calculations to be carried out using finite differences, we first show how a differencing step size $h \in (0, \gamma\|\nabla\phi(x_{i-1})\|]$ may be computationally determined without explicit knowledge of $\nabla\phi$.

In §2 we assumed that $\underline{f}(x)$ could be computed exactly or (in §2.3) to high relative precision. In this section, however, we assume only that $\underline{f} = \nabla\phi$ is computed approximately, probably by finite differences. Since the accuracy of a finite difference approximation varies with the differencing step size h, choosing $h \in (0,\gamma\|\nabla\phi(x)\|]$ is here a nontrivial task. This task may be carried out by means of the following

(1) Algorithm: [for determining $h \in (0,\gamma\|\nabla\phi(x)\|]$];

(1a) Let $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $x \in \mathbb{R}^n$, $h_1, \gamma_0 \in (0,\infty)$, $\theta \in (0,1)$ be given;

LOOP:

DO $i = 1$ TO ∞;

(1b) If $h_i \leq \gamma_0\left\|\left(\frac{\nabla\phi}{\nabla e_1}(x,h_i),\ldots,\frac{\nabla\phi}{\nabla e_n}(x,h_i)\right)\right\|$

(1c) Then stop and return $h = h_i$;

(1d) $h_{i+1} \leftarrow \theta h_i$;

END LOOP and Algorithm (1);

(2) Lemma: Let $\emptyset$, x, h_1, γ_0, θ be as in (1a). If $\emptyset'(x) \neq 0$, then Algorithm (1) halts. Moreover, for $\emptyset'(x) \neq 0$, if $K \in (0,\infty)$ and γ_0 satisfy

(3a) $$\max \|\emptyset''(\bar{B}(x,h_1))\| \leq K \quad \text{and}$$

(3b) $$\gamma_0 < (Kn)^{-1},$$

then the h returned in step (1c) satisfies

(4) $$h \leq \frac{\gamma_0 \|\nabla\emptyset(x)\|}{1-nK\gamma_0}.$$

Hence if $\gamma > 0$ with

(5a) $$\gamma_0 \leq \frac{\gamma}{1+nK\gamma},$$

then

(5b) $$h \leq \gamma\|\nabla\emptyset(x)\|.$$

If (3) holds and $\emptyset'(x) = 0$, then Algorithm (1) loops forever.

Proof: Step (1d) implies $h_i = \theta^{i-1}h_1$, so $h_i \to 0$ as $i \to \infty$; moreover, if $g_i = \left(\frac{\Delta\emptyset}{\Delta e_1}(x,h_i),\ldots,\frac{\Delta\emptyset}{\Delta e_n}(x,h_i)\right)$, then $\lim_{i\to\infty} g_i = \emptyset'(x)$, so (1b) must eventually be true whenever $\emptyset'(x) \neq 0$, i.e. Algorithm (1) halts if $\emptyset(x) \neq 0$.

Assume Algorithm (1) halts and let $h = h_i$ be the value returned. Assume further that (3) holds. By the MVT, $\|\emptyset'(y)-\emptyset'(z)\| \leq K\sqrt{n}\,\|y-z\|$ for all $y,z \in B(x,h_1)$, so, with g_i as above, (1.2.13) gives

$$\|g_i - \emptyset'(x)\| \le nKh_i \le nK\gamma_0\|g_i\|,$$

which implies

$$\|\nabla\emptyset(x)\| = \|\emptyset'(x)\| \ge (1-nK\gamma_0)\|g_i\|.$$

Since $nK\gamma_0 < 1$ by (3b),

$$h = h_i \le \gamma_0\|g_i\| \le \frac{\gamma_0\|\nabla\emptyset(x)\|}{1-nK\gamma_0},$$

which is (4); since $h_i = \theta^{i-1}h_1 > 0$, this also implies $\emptyset'(x) \ne 0$, so (1) does not halt if (3) holds and $\emptyset'(x) = 0$. If (3a) and (5a) hold, then $0 < \gamma_0 \le \gamma(1-nK\gamma_0)$, so $nK\gamma_0 < 1$ and $\gamma \ge \dfrac{\gamma_0}{1-nK\gamma_0}$, which together with (4) give (5b). ∎

Lemma (2) implies for any particular $\emptyset \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x \in \mathbb{R}^n$ with $\emptyset'(x) \ne 0$, Algorithm (1) halts and returns h satisfying (4) for all sufficiently small $\gamma_0 > 0$. Often for fixed h_1 and $x_0 \in \mathbb{R}^n$ there will exist $K \in (0,\infty)$ such that (3a) holds for all x in the level set $L(x_0) = \{x \in \mathbb{R}^n \mid \emptyset(x) \le \emptyset(x_0)\}$; in this case a sufficiently small and fixed γ_0 would suffice for all iterations of the descent algorithms we discuss. In principle there is a problem when $\emptyset'(x) = 0$, but in practice positive stopping tolerances would prevent infinite looping. Moreover, in practice h would probably be determined by another strategy, such as the one described in §3.6.

Now we state the general form of the proposed minimization algorithms near a nonsingular minimizer x^*. We have in mind that the approximate derivatives called for will be computed by finite differences, so that

$\xi = \kappa h^{\mu}$, where h might be determined by Algorithm (1).

(6) <u>Algorithm</u>: [general form of proposed minimization algorithms near x^*];

(6a) Let $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $x_0 \in \mathbb{R}^n$, $\Xi : \mathbb{R}^n \to [0,\infty]$, and the constants $c_0, c_1, \delta, \epsilon_5, \epsilon_6, \epsilon_7, \epsilon_8$ be given;

OUTERLOOP:

DO $i = 1$ TO ∞;

(6b) If $\phi'(x_{i-1}) = 0$ then stop and let $x_j = x_{i-1}$ for all $j \geq i$;

(6c) $y_0 \leftarrow x_{i-1}$;

(6d) $\xi \leftarrow \Xi(y_0)$;

INNERLOOP:

DO $k = 1$ TO n;

(6e(i)) Let $W_{k-1} = [w_1, \ldots, w_{k-1}]$,

(6e(ii)) $P_{k-1} = I - W_{k-1}W_{k-1}^{+}$, and

(6f) $Q_{k-1} = [q_1, \ldots, q_{k-1}]$;

(6g) Choose $p_k \neq 0$ with $\|p_k - P_{k-1}p_k\| \leq \delta\xi\|p_k\|$;

(6h) $\tilde{q}_k \leftarrow (I - Q_{k-1}Q_{k-1}^{+})p_k$;

(6i) $q_k \leftarrow \tilde{q}_k / \|\tilde{q}_k\|$;

(6j) Find $\bar{f}_k$ with $\left|\frac{\partial\phi}{\partial q_k}(y_{k-1}) - \bar{f}_k\right| \leq \epsilon_5\xi\|\phi'(y_0)\|$;

(6k) Find π_k with $\left|\frac{\partial^2\phi}{\partial q_k \partial p_k}(y_{k-1}) - \pi_k\right| \leq \epsilon_6\xi\|p_k\|$;

(6ℓ) Choose $y_k \in \bar{B}\left(y_{k-1} - \frac{\bar{f}_k}{\pi_k} p_k,\ \epsilon_7 \xi \|\phi'(y_0)\|\right)$;

(6m) Find a basis $u_k^{(k)}, \ldots, u_n^{(k)}$ for $\mathrm{Span}\{w_1, \ldots, w_{k-1}\}^{\perp}$ with

(6m(i)) $\chi(U_k) \le c_0 \|U_k\|$ and

(6m(ii)) $\|u_j^{(k)}\| \le c_1$, where

(6m(iii)) $U_k = [u_k^{(k)}, \ldots, u_n^{(k)}]$;

(6n) For $k \le j \le n$ find $\omega_j^{(k)}$ with

$$\left|\omega_j^{(k)} - \frac{\partial^2 \phi}{\partial q_k \partial u_j^{(k)}} (y_{k-1})\right| \le \epsilon_8 \xi;$$

(6o) $$w_k \leftarrow U_k (U_k^T U_k)^{-1} \begin{pmatrix} \omega_k^{(k)} \\ \vdots \\ \omega_n^{(k)} \end{pmatrix};$$

END INNERLOOP;

(6p) $x_i \leftarrow y_n$;

END OUTERLOOP and Algorithm (6);

The calculation called for in step (6h) may be carried out conveniently by means of Householder or Givens transformations. If Householder transformations are used, then $n^2/2 + \mathcal{O}(n)$ storage locations are required in general, whereas Givens transformations would need $n^2 + \mathcal{O}(n)$ storage locations and roughly twice as much arithmetic for problems without special sparseness structure.

The specific algorithms we propose all generate the basis $u_{k+1}^{(k+1)},\ldots,u_n^{(k+1)}$ at the end of the $k^{\underline{th}}$ minor iteration, using $\omega_k^{(k)},\ldots,\omega_n^{(k)}$ to do so, then choose $p_{k+1} \in \mathrm{Span}\{u_{k+1}^{(k+1)},\ldots,u_n^{(k+1)}\}$ at the start of the next minor iteration. Under exact arithmetic this choice results in (6g) holding with $\delta = 0$. Preliminary computational results suggest that it matters little just how $p_k \in \mathrm{Span}\{u_k^{(k)},\ldots,u_n^{(k)}\}$ is chosen.

Assume x^* is a nonsingular minimizer of $\emptyset$ and that $\emptyset''$ satisfies a Lipschitz condition of order $\mu \in (0,1]$ near x^*. Suppose $\emptyset'(y_0) \neq 0$ and that $h \in (0,\gamma\|\emptyset'(y_0)\|]$. Lemma (1.2.11) shows that one of the central difference operators defined in (1.2.3,7,8) may be used to carry out the calculation called for in step (6j) with $\xi = \kappa h^{\mu}$ and $\epsilon_5 \leq 2^{\mu}\gamma$; this lemma shows that the operators defined in (1.2.4,5) may be used to compute π_k and $\omega_j^{(k)}$ in steps (6k,n) in a similar way.

It is possible to replace q_k by p_k in Algorithm (6). This may allow saving a few function evaluations because the difference operators defined in (1.2.6,9,10) may then be used to compute π_k. If p_k replaces q_k in steps (6j,k,n), then q_k need not be explicitly calculated and the housekeeping work in Algorithm (6) may thus be reduced, albeit with a possible small loss of accuracy. Such replacement is justified in Lemma (9), the proof of which requires the following

(7) Lemma: If $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ is positive definite and $p \in \mathbb{R}^n - \{0\}$, then

$$(8)\qquad \frac{p^TAp}{\|p\|\,\|Ap\|} \geq \frac{2\sqrt{\chi(A)}}{\chi(A)+1}\ .$$

Proof: Let $d_1,\ldots,d_n > 0$ be the eigenvalues of A. By an orthogonal change of basis we may replace A by $D = \mathrm{diag}(d_1,\ldots,d_n)$. Let $\psi\colon (\mathbb{R}^n - \{0\}) \to \mathbb{R}$ be given by

$$\psi(x) \equiv \frac{x^TDx}{\|x\|\,\|Dx\|}\ .$$

It suffices to show that $\min \psi(\mathbb{R}^n - \{0\}) = \mathrm{RHS}(8) = 2\sqrt{\chi(D)}\,/(\chi(D)+1)$.

Since $\psi(x) = \psi(x/\|x\|)$, we have $\psi(\mathbb{R}^n - \{0\}) = \psi(S)$, where S is the unit sphere $S = \{y \in \mathbb{R}^n : \|y\| = 1\}$ of $\mathbb{R}^n$. Because S is compact, ψ attains its minimum, say at $z \in S$. Now

$$\nabla\psi(x) = \frac{2\|x\|^2\|Dx\|^2Dx - (x^TDx)[\|Dx\|^2x + \|x\|^2D^2x]}{\|x\|^3\|Dx\|^3}\ ,$$

so $\nabla\psi(z) = 0$ and $\|z\| = 1$ imply

$$2\|Dz\|^2Dz - (z^TDz)[\|Dz\|^2z + D^2z] = 0,$$

whence z, Dz, and D^2z are linearly dependent. WLOG $z = (z_1,\ldots,z_n)^T$ has $z_3 = \ldots = z_n = 0$, because if z_j were nonzero for three or more distinct values of d_j, then z, Dz, and D^2z would be linearly independent; and if, say, $z_2z_3 \neq 0$ with $d_2 = d_3$, then we could rotate the coordinate system to achieve $z_3 = 0$. Let $T\colon [0,1] \to \mathbb{R}$ be given by

$$T(t) = \frac{(d_1-d_2)t+d_2}{\sqrt{(d_1^2-d_2^2)t+d_2^2}}\ .$$

Since $z_1^2 + z_2^2 = \|z\|^2 = 1$,

$$\psi(z) = \frac{d_1 z_1^2 + d_2 z_2^2}{\sqrt{d_1^2 z_1^2 + d_2^2 z_2^2}} = T(z_1^2),$$

so $\min \psi(\mathbb{R}^n - \{0\}) = \min T([0,1])$. Now

$$2T'(t) = \frac{2(d_1-d_2)[(d_1^2-d_2^2)t+d_2^2] - (d_1^2-d_2^2)[(d_1-d_2)t+d_2]}{\left[(d_1^2-d_2^2)t+d_2^2\right]^{3/2}}$$

$$= (d_2-d_1)^2[(d_1+d_2)t - d_2]\left[(d_1^2-d_2^2)t + d_2^2\right]^{-3/2},$$

so T is minimized over $[0,1]$ at $t^* = \dfrac{d_2}{d_1+d_2}$. Thus

$$\min \psi(\mathbb{R}^n - \{0\}) = T(t^*) = \frac{2\sqrt{d_1 d_2}}{d_1+d_2} = \frac{2\sqrt{d_1/d_2}}{(d_1/d_2)+1}.$$ WLOG $d_1 \geq d_2$. Since

$\Phi(t) \equiv \dfrac{2t^{1/2}}{t+1}$ has $\Phi'(t) = \dfrac{1-t}{t^{1/2}(t+1)^2} < 0$ for $t > 1$, we must have

$d_1/d_2 = \chi(D) = \chi(A)$, whence follows (8). ∎

The next lemma is the main result of this subsection: it proves local convergence of Algorithm (6) and gives a lower bound on the rate of Q-convergence. Its proof shows that q_k may be replaced by $p_k/\|\tilde{q}_k\|$ in step (6j), (6k), and/or (6n).

(9) Lemma: Suppose $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\phi'(x^*) = 0$ with $\phi''(x^*)$ positive definite. For each choice of the constants $c_0, c_1 \in (0,\infty)$, $\epsilon_5, \epsilon_6, \epsilon_7, \epsilon_8 \in [0,\infty)$, the conclusions of Theorem (2.1.62) with $\underline{f} = \nabla\phi$ hold for Algorithm (6). The same remains true if q_k is

replaced by $p_k/\|\tilde{q}_k\|$ in one or more of steps (6j,k,n).

<u>Proof</u>: It suffices to show the existence of $\epsilon_1, \epsilon_2, \epsilon_4 \in (0,\infty)$ and $\eta \in (0,1)$ such that (2.1.54f(ii),g,h,j) hold near x^* for all small enough ξ with $\underline{f} = \nabla\phi$ and $f_k = \frac{\partial\phi}{\partial q_k}$. Note that if steps (6k) and (6n) remain unchanged, then we may take $\epsilon_1 = \epsilon_6$ and $\epsilon_4 = \epsilon_8$. We shall show inductively that there exist $\rho^{(k)}$, $\xi_0^{(k)}$, $\epsilon_1^{(k)}$, $\epsilon_2^{(k)}$, $\epsilon_4^{(k)}$, and $\eta^{(k)}$ such that for $1 \leq \nu \leq k$,

(10a) $$|p_\nu \phi''(y_{\nu-1})q_\nu| \geq \eta^{(k)}\|P_{k-1}\phi''(y_{\nu-1})q_\nu\|\|p_\nu\|,$$

(10b) $$|\pi_\nu - p_\nu\phi''(y_{\nu-1})q_\nu| \leq \epsilon_1^{(k)}\xi\|p_\nu\|,$$

(10c) $$y_\nu \in \bar{B}\left(y_{\nu-1} - \left[\frac{\phi'(y_{\nu-1})q_\nu}{\pi_\nu}\right]p_\nu,\ \ \epsilon_2^{(k)}\xi\|\phi'(y_0)\|\right), \text{ and}$$

(10d) $$|\omega_j^{(\nu)} - q_\nu^T\phi''(y_{\nu-1})u_j^{(\nu)}| \leq \epsilon_4^{(k)}\xi, \qquad \nu \leq j \leq n,$$

whenever $y_0,\ldots,y_{\nu-1} \in B(x^*,\rho^{(k)})$ and $\xi \leq \xi_0^{(k)}$.

For $k = 1$, assume WLOG that $\|p_1\| = 1$, whence $q_1 = p_1$. Let $\rho^{(1)} > 0$ be so small that $\frac{2\sqrt{\chi(\phi''(x))}}{\chi(\phi''(x))+1} \geq \frac{\sqrt{\chi(\phi''(x^*))}}{\chi(\phi''(x^*))+1} \equiv \eta^{(1)}$ and $\|\phi''(x)^{-1}\| \leq 2\|\phi''(x^*)^{-1}\|$ whenever $x \in B(x^*,\rho^{(1)})$. Lemma (7) implies that (10a) holds for $k = \nu = 1$ if $y_0 \in B(x^*,\rho^{(1)})$. Steps (6k) and (6n) assure that (10b,d) hold for $k=1$ with $\epsilon_1^{(1)} = \epsilon_6$ and $\epsilon_4^{(1)} = \epsilon_8$. Let $\xi_0^{(1)} = \frac{\eta^{(1)}\min\{1,1/\epsilon_6\}}{4\|\phi''(x^*)^{-1}\|}$: since $P_0 = I$ and $\|\phi''(y_0)q_1\| \geq 1/\|\phi''(y_0)^{-1}\|$, (10a,b) imply $|\pi_1| \geq \xi_0^{(1)}\|p_1\|$ whenever $y_0 \in B(x^*,\rho^{(1)})$ and $\xi \leq \xi_0^{(1)}$. Under these conditions, (6ℓ,j) imply

$$\left\|y_1 - \left(y_0 - \frac{\emptyset'(y_0)q_1}{\pi_1}\right)\right\| \leq \epsilon_7\xi\|\emptyset'(y_0)\| + \frac{|\emptyset'(y_0)q_1-\bar{f}_1|}{|\pi_1|}\|p_1\|$$

$$\leq (\epsilon_7 + \epsilon_5/\xi_0^{(1)})\xi\|\emptyset'(y_0)\|.$$

Hence (10c) holds with $\epsilon_2^{(1)} = \epsilon_7 + \epsilon_5/\xi_0^{(1)}$.

Suppose inductively that $2 \leq k \leq n$ and there exist $\epsilon_1^{(k-1)}$, $\epsilon_2^{(k-1)}$, $\epsilon_4^{(k-1)}$, $\eta^{(k-1)}$, $\rho^{(k-1)}$, $\xi_0^{(k-1)}$ that (10) holds with k replaced by $k-1$ whenever $y_0,\ldots,y_{k-2} \in B(x^*,\rho^{(k-1)})$ and $\xi \leq \xi_0^{(k-1)}$. Depending on which assertion of Theorem (2.1.62) we are proving, we have $\mu = 0$ or $\mu \in (0,1]$ and $\rho_2 \in (0,\infty]$, $\kappa \in [0,\infty)$ such that

(11) $$\|\emptyset''(x)-\emptyset''(y)\| \leq \kappa\|x-y\|^{\mu} \quad \text{for all } x,y \in B(x^*,\rho_2).$$

If $\mu = 0$, we may choose $\rho_2 > 0$ and let

(12) $$\kappa = \kappa(\rho_2) = \max\{\|\emptyset''(x)-\emptyset''(y)\|: x,y \in \bar{B}(x^*,\rho_2)\};$$

the continuity of $\emptyset''$ implies that we may make κ as small as we please by choosing ρ_2 properly. In this case, let the following be carried out for a particular choice of κ, ρ_2 such that (11) holds. By Lemma (2.1.20) we may assume that $\xi_0^{(k-1)}$ and $\rho_1 \equiv \rho_0^{(k-1)}$ are so small that there exists $K \in (1,\infty)$ such that $\|y_j-x^*\| \leq K\|y_0-x^*\|$ whenever $y_0 \in B(x^*,\rho_1)$ and $\xi \leq \xi_0^{(k-1)}$. Assuming WLOG that $\kappa(K\rho_1) \leq \xi_0^{(k-1)}$, where $\kappa(\rho)$ is defined in (12), for $\mu = 0$ we may assume WLOG that $\Xi(x) = +\infty$ for

$\|x-x^*\| \geq \rho_1$ and $\Xi(x) \geq \kappa(K\|x-x^*\|)$ for $x \in B(x^*,\rho_1)$. Similarly, for $\mu > 0$, we may assume WLOG that

$\Xi(x) \geq \kappa(2\|x-x^*\|K)^{\mu}$ for $x \in B(x^*,\rho_1)$ and $\Xi(x) = +\infty$ if $\|x-x^*\| \geq \rho_1$ or $\Xi(x) > \xi_0^{(k-1)}$. In both cases we may therefore assume (11) holds and

$$\kappa\|y_{k-1}-y_j\|^{\mu} \leq \xi, \quad 0 \leq j \leq k-2, \tag{13a}$$

whenever $y_0,\ldots,y_{k-1} \in B(x^*,\rho^{(k-1)})$. Since $K > 1$, we may also assume

$$\kappa\|y_j-x^*\|^{\mu} \leq \xi, \quad 0 \leq j \leq k-1. \tag{13b}$$

First we show that if ξ is small enough, then

$$\|\tilde{q}_k\| \geq \frac{\|p_k\|}{2(1+\chi(\emptyset''(x^*))} . \tag{14}$$

For the moment, let

$$A = \emptyset''(x^*). \tag{15}$$

We need to bound $|p_k^T A q_j|$, $j < k$. Now

$$p_k^T A q_j = p_k^T w_j + p_k^T(P_{j-1}\emptyset''(y_{j-1})q_j - w_j) + (p_k - P_{j-1}p_k)^T \emptyset''(y_{j-1})q_j + \tag{16}$$

$$+ p_k^T[\emptyset''(x^*)-\emptyset''(y_{j-1})]q_j.$$

Assume $y_0,\ldots,y_{k-1} \in B(x^*,\rho^{(k-1)})$ with $\xi \leq \xi_0^{(k-1)}$, and let

$$K_2 = \max\|\emptyset''(\bar{B}(x^*,\rho^{(k-1)}))\|. \tag{17}$$

Since $\|q_j\| = \|p_{j-1}\| = 1$, $1 \le j \le k-1$, and since (10d), (6m(i)), and Lemma (2.1.55) combine to say that

$$\|w_j - P_{j-1}\phi''(y_{j-1})q_j\| \le \epsilon_3^{(k)}\xi, \quad 1 \le j \le k-1, \tag{18}$$

with $\epsilon_3^{(k)} = c_0\epsilon_4^{(k-1)}\sqrt{n}$, we have

$$\|w_j\| \le K_2 + \epsilon_3^{(k)}\xi .$$

Since (6g) implies $|p_k^T w_j| \le \delta\xi\|p_k\|\|w_j\|$, we thus have

$$|p_k^T w_j| \le K_3\xi\|p_k\|, \qquad 1 \le j \le k-1 \tag{19}$$

where $K_3 = (K_2+\epsilon_3^{(k)}\xi_0^{(k-1)})\delta$. Hence, using (16), (19), (18), (6g), (17), (6i), (11), and (13b) we find

$$|p_k^T A q_j| \le K_1\xi\|p_k\|, \qquad 1 \le j \le k-1, \tag{20}$$

where $K_1 = K_3 + \epsilon_3^{(k)} + \delta K_2 + 1$.

Note that $q_1,\ldots,q_{k-1}$ are orthonormal. Let $z = Q_{k-1}^T p_k$, so that

$$p_k = \tilde{q}_k + Q_{k-1}z, \tag{21}$$

where Q_{k-1} and $\tilde{q}_k$ come from (6f,h). Let

$$v = Q_{k-1}^T A p_k = B\tilde{q}_k + BQ_{k-1}z, \tag{22a}$$

where

$$B = Q_{k-1}^T A. \tag{22b}$$

Note that (20) implies

$$\|v\| \leq K_1 \xi \|p_k\| \sqrt{k-1}\ . \tag{23}$$

Let $\sigma = \|B\|/\chi(B)$ be the least singular value of B. Using (21), we find

$$\|BQ_{k-1}z\| \geq \sigma\|Q_{k-1}z\| \geq \sigma(\|p_k\| - \|\tilde{q}_k\|).$$

From (22a) we thus have

$$\|B\|\|\tilde{q}_k\| \geq \|B\tilde{q}_k\| \geq \|BQ_{k-1}z\| - \|v\|$$

$$\geq \sigma(\|p_k\| - \|\tilde{q}_k\|) - \|v\|, \quad \text{whence}$$

$$(\|B\|+\sigma)\|\tilde{q}_k\| \geq \sigma\|p_k\| - \|v\|.$$

Now $\|Q_{k-1}x\| = \|x\|$ for all $x \in \mathbb{R}^{k-1}$, so $\chi(A) \geq \chi(B)$ and $\|A\|/\chi(A) \leq \|B\|/\chi(B) = \sigma$. Hence

$$\|\tilde{q}_k\| \geq \frac{\sigma\|p_k\| - \|v\|}{\sigma + \|B\|} = \frac{\|p_k\| - \|v\|/\sigma}{1+\chi(B)} \geq \frac{\|p_k\| - \|v\|\chi(A)/\|A\|}{1+\chi(A)},$$

which, with (23), implies

$$\|\tilde{q}_k\| \geq \left[\frac{1 - K_1\xi\sqrt{k-1}\ \chi(A)/\|A\|}{1+\chi(A)}\right]\|p_k\|.$$

Recalling (15), we thus see that (14) holds if

$$\xi \leq \|\emptyset''(x^*)\|/[2K_1\chi(\emptyset''(x^*))\sqrt{k-1}\ ]. \tag{24}$$

We next show that

(25a) $$|q_j^T \emptyset''(y_{k-1})u| \le (1+\epsilon_3^{(k)})\xi, \qquad 1 \le j \le k-1,$$

whenever

(25b) $$u \in \operatorname{Span}\{w_1,\ldots,w_{k-1}\}^{\perp}.$$

Indeed, (25b) and (18) imply $u = P_{k-1}u$ and

$$|u^T \emptyset''(y_{j-1})q_j| \le \epsilon_3^{(k)}\xi\|u\|,$$

which, together with (11) and (13a), implies (25a).

Let $z = Q_{k-1}^T p_k$, so that (21) holds. WLOG $\|\tilde{q}_k\| = 1$, whence $q_k = \tilde{q}_k$ and (21) becomes

(26) $$p_k = q_k + Q_{k-1}z.$$

From (26), (25), (14), the inequality $\|z\| \le \|p_k\|$, and the assumption $\|\tilde{q}_k\| = 1$ we obtain

(27a) $$|(q_k-p_k)^T \emptyset''(y_{k-1})u| \le K_4\xi\|u\|,$$

whenever

(27b) $$u \in \operatorname{Span}\{w_1,\ldots,w_{k-1}\}^{\perp},$$

where $K_4 = 2(1+\epsilon_3^{(k)})\sqrt{k-1}\,(1+\chi(\emptyset''(x^*)))$. In particular, if p_k replaces q_k in (6n), then (27) and (6m(ii)) imply that (10d) holds with $\epsilon_4^{(k)} = \min\{\epsilon_4^{(k-1)}, c_1K_4 + \epsilon_8\}$.

Let $\underline{p} = P_{k-1}p_k$ and $\overline{p} = W_{k-1}W_{k-1}^{+}p_k = p_k - \underline{p}$. By (6g),

$\|\bar{p}\| \le \delta\xi\|p_k\|$. Since

$$|(p_k-q_k)^T\phi''(y_{k-1})p_k| \le |(p_k-q_k)^T\phi''(y_{k-1})\underline{p}| + \|p_k-q_k\|\|\phi''(y_{k-1})\|\|\bar{p}\|$$

and $\|p_k-q_k\| \le \|p_k\|$, we find using (17), (27), and (14) that

(28) $$|(p_k-q_k)^T\phi''(y_{k-1})p_k| \le K_5\xi\|p_k\|,$$

where $K_5 = K_4 + 2K_2\delta[1+\chi(\phi''(x^*))]$. In particular, if p_k replaces q_k in (6k), then (10b) holds with $\epsilon_1^{(k)} = \max\{\epsilon_1^{(k-1)}, \epsilon_6 + K_5\}$.

Because $\rho^{(k-1)} \le \rho^{(1)}$ and $y_0,\ldots,y_{k-1}$ $B(x^*,\rho^{(k-1)})$, Lemma (7) and the choice of $\eta^{(1)}$ imply

(29) $$|p_k^T\phi''(y_{k-1})p_k| \ge \eta^{(1)}\|p_k\|\|\phi''(y_{k-1})p_k\|.$$

Now $1 = \|q_k\| = \|\tilde{q}_k\| \le \|p_k\|$ implies

$$\|\phi''(y_{k-1})p_k\| \ge \|\phi''(y_{k-1})q_k\|/\chi(\phi''(y_{k-1})).$$

Since $\|P_{k-1}\| = 1$, we have by (28) and (29) that

(30) $$|q_k^T\phi''(y_{k-1})p_k| \ge \left[\left(\|P_{k-1}\phi''(y_{k-1})q_k\|\eta^{(1)}/\chi(\phi''(y_{k-1}))\right)-K_5\xi\right]\|p_k\|.$$

Lemma (2.1.15) assures the existence of $\xi_0 = \xi_0^{(k)}$ and $\rho^{(k)} \le \rho^{(k-1)}$ with $\xi_0^{(k)} \le \min\{\xi_0^{(k-1)}, \text{RHS}(24)\}$ such that

$$\|P_{k-1}\phi''(y_{k-1})q_k\| \ge \frac{1}{2\|\phi''(x^*)^{-1}\|}$$

whenever $\xi \le \xi_0^{(k)}$ and $y_0,\ldots,y_{k-1} \in B(x^*,\rho^{(k)})$. WLOG $\rho^{(k)}$ is small enough that $\chi(\emptyset''(x)) \le 2\chi(\emptyset''(x^*))$ for all $x \in B(x^*,\rho^{(k)})$. Furthermore, we may assume that $\xi_0^{(k)}$ satisfies

$$\xi_0^{(k)} \le \eta^{(k)} \equiv \min\left\{\frac{\eta^{(1)}}{8K_5\chi(\emptyset''(x^*))\|\emptyset''(x^*)^{-1}\|},\ \eta^{(k-1)}\right\}.$$

From (30) we see that (10a) is satisfied whenever $\xi \le \xi_0^{(k)}$ and $y_0,\ldots,y_{k-1} \in B(x^*,\rho^{(k)})$.

As in the proof of Lemma (2.1.40), we find (in view of (13a)) that

$f_j(y_{k-1}) \equiv \frac{\partial\emptyset}{\partial q_j}(y_{k-1}) = \emptyset'(y_{k-1})q_j = \mathcal{O}(\xi\|\emptyset'(y_0)\|)$ for $1 \le j \le k-1$. From (26), (14), and the assumption $\|\tilde{q}_k\| = 1$ we thus have

$\emptyset''(y_{k-1})p_k = \emptyset'(y_{k-1})q_k + \mathcal{O}(\xi\|\emptyset(y_0)\|)$. Hence, by reasoning similar to that for $k=1$, we may assume $\rho^{(k)}$ and $\xi_0^{(k)}$ to be small enough that (10c) holds for some $\epsilon_2^{(k)} > 0$ whenever $\xi \le \xi_0^{(k)}$ and $y_0,\ldots,y_{k-1} \in B(x^*,\rho^{(k)})$, whether or not $p_k = p_k/\|\tilde{q}_k\|$ replaces q_k in step (6j). This completes the induction.

With or without $p_k/\|\tilde{q}_k\|$ replacing q_k in one or more of steps (6j,k,n), Algorithm (6) thus satisfies the conditions of Algorithm (2.1.54) with $\underline{f} = \nabla\emptyset$ whenever $y_0,\ldots,y_n \in B(x^*,\rho^{(n)})$ and $\xi \le \xi_0^{(n)}$. Since we may see Lemma (2.1.20) to modify Ξ in such a way that Algorithm (6) behaves like Algorithm (2.1.54) everywhere (i.e. $\Xi(x) = +\infty$ when $\|x-x^*\|$ is too large), Theorem (2.1.62) applies. ∎

Notice that if $\bar{f}_k$ and π_k are scaled by the same nonzero factor, then the performance of Algorithm (6) remains unchanged. The same holds true if $\omega_k^{(k)},\ldots,\omega_n^{(k)}$ are so scaled. Thus the scale factor $1/\|\tilde{q}_k\|$ may be omitted from p_k if p_k replaces q_k in both (6j) and (6k) or in (6n) for all j, $k \le j \le n$. Now if $\bar{f}_k = \frac{\tilde{\Delta}\phi}{\Delta p_k}(y_{k-1},h)$ and $\pi_k = \frac{\tilde{\Delta}^2\phi}{\Delta p_k^2}(y_{k-1},h)$ (see (1.2.3,6)), then only the values $\phi(y_{k-1})$ and $\phi(y_{k-1} \pm hp_k)$ are needed to compute $\bar{f}_k$ and π_k. Those of the implementations suggested in §3.5 which choose $p_k = u_k^{(k)}$ may do this and then let $\omega_k^{(k)} = \pi_k$. If they do choose to use q_k when computing $\omega_{k+1}^{(k)},\ldots,\omega_n^{(k)}$, then scaling by $\|\tilde{q}_k\|$ is important.

In cases where $\phi''(x^*)$ is ill conditioned, using q_k rather than p_k might lead to slightly more accurate approximations--some of the error bounds in the proof of Lemma (9) are smaller for q_k than for $p_k/\|\tilde{q}_k\|$. On the other hand, as just seen, using p_k instead of q_k can save one or two function evaluations. It might therefore sometimes be desirable to use a mixed strategy like the one indicated above in which $p_k/\|\tilde{q}_k\|$ replaces q_k in some but not all of the calculations. Preliminary computational results make no strong case one way or the other, though replacing q_k by p_k when calculating $\bar{f}_k$ and π_k does appear to have some advantage.

If $\omega_k^{(k)},\ldots,\omega_n^{(k)}$ are not computed until after y_k has been found, as indicated in Algorithm (6), then we may contemplate replacing y_{k-1} by y_k in step (6n). Such a modification does not affect Lemma (9):

(31) Lemma: Lemma (9) remains valid if y_{k-1} is replaced by y_k in step (6n), i.e. if (6n) is changed to

(32) For $k \le j \le n$, find $\omega_j^{(k)}$ with

$$|\omega_j^{(k)} - \frac{\partial^2 \phi}{\partial q_k \partial u_j^{(k)}}(y_k)| \le \epsilon_8 \xi .$$

Proof sketch: Reasoning as in the proof of Lemma (9), we find $\|y_k - y_{k-1}\| = \mathcal{O}(\|y_0 - x^*\|)$. Hence if $\omega_j^{(k)}$ satisfies (32), then (11) and (13b) imply that (6n) holds with a larger value of ϵ_8 whenever $\|y_k - x^*\|$ is not too large. ∎

During preliminary testing, (32) led to sometimes better, sometimes worse performance than (6n).

3.3 ψ_1: The Inner Loop Plus Tests

In this subsection we construct an algorithmically defined function $\psi_1 : \mathbb{R}^n \to \mathbb{R}^n$ consisting of most of the inner loop of Algorithm (3.2.6) together with some tests. We show that if x^* is a nonsingular minimizer of ϕ and if the constants involved in the tests are properly chosen, then ψ_1 combined with an appropriate calling program reduces to Algorithm (3.2.6) near x^*. The tests assure that the minor iterates $y_1, \ldots, y_k$ satisfy $\phi(y_1) \ge \phi(y_2) \ge \ldots \ge \phi(y_k)$. Indeed, minor iterate y_k may be called "successful" if $\phi(y_k)$ exceeds neither $\phi(y_{k-1})$ nor a certain computed bound. With this terminology, the calling program is expected to construct the first successful iterate y_1 and pass it to ψ_1 as a parameter. The value which ψ_1 returns is the last successful minor iterate y_k, $k \le n$, which ψ_1 has found. Thus $\phi(\psi_1(y_1)) \le \phi(y_1)$ for any $y_1 \in \mathbb{R}^n$.

The following lemma suggests how we may compute an upper bound on $\phi(y_k) - \phi(y_{k-1})$ near x^*.

(1) Lemma: Assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\phi'(x^*) = 0$ with $\phi''(x^*)$ positive definite. Assume further that there exist $\mu \in [0,1]$, $\rho_2 \in (0,\infty]$, $\kappa \in [0,\infty)$ such that

(2) $$\|\phi''(x) - \phi''(y)\| \le \kappa\|x-y\|^\mu \quad \text{for all } x,y \in B(x^*,\rho_2).$$

For each choice of the constants $c_0, c_1 \in (0,\infty)$, $\epsilon_5,\epsilon_6,\epsilon_7,\epsilon_8 \in [0,\infty)$ mentioned in Algorithm (3.2.6) and each $\Xi: \mathbb{R}^n \to [0,\infty]$ such that

(3) $$\Xi(x) = o(\|\phi'(x)\|^\mu) \quad \text{for all } x \in B(x^*,\rho_2)$$

there exist $\rho_3 \in (0,\rho_2)$ and $K \in (0,\infty)$ such that if $y_0 \in B(x^*,\rho_3)$ and $y_1,\dots,y_k$ are generated from y_0 by the inner loop of Algorithm (3.2.6) with $\xi = \Xi(y_0)$ and step (3.2.6ℓ) changed in the k^{th} minor iteration to

(4) $$\text{"Choose } y_k \in B(y_{k-1} - \frac{\bar{f}_k}{\pi_k} p_k,\ \epsilon_7\xi\|\phi'(y_0)\|^{1+\mu});\text{"}$$

then

(5) $$\phi(y_k) \le \phi(y_{k-1}) - \frac{1}{2}\frac{(\bar{f}_k)^2}{\pi_k}\|p_k\| + K\|\phi'(y_0)\|^{1+\mu}\max\{|\bar{f}_k|, \|\phi'(y_0)\|^{1+\mu}\}.$$

Proof: First we show that

(6) $$|\phi'(y+tp)p - \phi'(y)p - tp^T\phi''(y)p| \le \frac{\kappa|t|^{1+\mu}\|p\|^{2+\mu}}{1+\mu}$$

whenever $y, y+tp \in B(x^*,\rho_2)$, $t \in \mathbb{R}$, $p \in \mathbb{R}^n$. Indeed, we have

$$\phi'(y+tp)p - \phi'(y)p - tp^T\phi''(y)p = \int_0^1 tp^T[\phi''(y+\tau tp) - \phi''(y)]p\,d\tau,$$

so (2) implies

$$\text{LHS}(6) \le \kappa|t|^{1+\mu}\|p\|^{2+\mu}\int_0^1 \tau^\mu d\tau = \frac{\kappa|t|^{1+\mu}\|p\|^{2+\mu}}{1+\mu} = \text{RHS}(6).$$

Assume now that p_k, $\bar{f}_k$, π_k, and y_k are generated from y_{k-1} in the inner loop of Algorithm (3.2.6) with $y_{k-1}, y_k \in B(x^*,\rho_2)$. Let $\underline{y}_k = y_{k-1} - \left(\frac{\bar{f}_k}{\pi_k}\right)p_k$ and assume also that $\underline{y}_k \in B(x^*,\rho_2)$. Using (6), we find

$$\phi(\underline{y}_k) = \phi(y_{k-1}) + \int_0^{-\bar{f}_k/\pi_k} \phi'(y_{k-1}+tp_k)p_k\,dt$$

$$\le \phi(y_{k-1}) + \int_0^{-\bar{f}_k/\pi_k} [\phi'(y_{k-1})p_k + tp_k^T\phi''(y_{k-1})p_k]dt +$$

$$+ \frac{\kappa\|p_k\|^{2+\mu}}{1+\mu}\int_0^{|\bar{f}_k/\pi_k|} t^{1+\mu}dt$$

$$= \phi(y_{k-1}) - \left(\frac{\bar{f}_k}{\pi_k}\right)\phi'(y_{k-1})p_k + \frac{1}{2}\left(\frac{\bar{f}_k}{\pi_k}\right)^2 p_k^T\phi''(y_{k-1})p_k +$$

$$+ \frac{\kappa}{(1+\mu)(2+\mu)}\left\|\left(\frac{\bar{f}_k}{\pi_k}\right)p_k\right\|^{2+\mu},$$

whence

$$\text{(7)}\quad \phi(y_k) \le \phi(y_{k-1}) - \frac{1}{2}\|p_k\|\frac{(\bar{f}_k)^2}{\pi_k} + |\phi(\underline{y}_k) - \phi(y_k)| + \left[\frac{\bar{f}_k}{\pi_k}\left(\|p_k\|\bar{f}_k - \phi'(y_{k-1})p_k\right)\right.$$
$$\left. + \frac{1}{2}\left(\frac{\bar{f}_k}{\pi_k}\right)^2\left(p_k^T\phi''(y_{k-1})p_k - \|p_k\|\pi_k\right) + \frac{\kappa}{(1+\mu)(2+\mu)}\left\|\left(\frac{\bar{f}_k}{\pi_k}\right)p_k\right\|^{2+\mu}\right].$$

From (3),(4), Lemma (2.1.20), and the proof of Lemma (3.2.9) we see that there exists $\rho_3 \in (0,\rho_2)$ such that if $y_0 \in B(x^*,\rho_3)$ then $y_0, \underline{y}_1, y_1, \ldots, \underline{y}_n, y_n \in B(x^*,\rho_2)$, $|\bar{f}_k| = \mathcal{O}(\|\phi'(y_0)\|)$, $|\bar{f}_k\|p_k\| - \phi'(y_{k-1})p_k| = \mathcal{O}(\|p_k\|\|\phi'(y_0)\|^{1+\mu})$, $\left\|\frac{p_k}{\pi_k}\right\|$ is bounded above, $|p_k^T\phi''(y_{k-1})p_k - \|p_k\|\pi_k| = \mathcal{O}(\|p_k\|^2\|\phi'(y_0)\|^{\mu})$, $\max\{\|\phi'(y_k+t(\underline{y}_k-y_k))\|: 0 \le t \le 1\} = \mathcal{O}(\|\phi'(y_0)\|)$, and $\|\underline{y}_k-y_k\| = \mathcal{O}(\|\phi'(y_0)\|^{1+2\mu})$. Thus $|\phi(\underline{y}_k)-\phi(y_k)| = \mathcal{O}(\|\phi'(y_0)\|^{2+2\mu})$ and the terms in square brackets in (7) are bounded by $\|\phi'(y_0)\|^{1+\mu}\,\mathcal{O}(\max\{|\bar{f}_k|, \|\phi'(y_0)\|^{1+\mu}\})$, so there exists $K \in (0,\infty)$ such that (5) holds. ∎

The inner loop algorithm ψ_1 stated below uses the higher precision arithmetic called for in (4) to compute a candidate $\bar{y}_k$ for y_k. Near x^*, if $\phi(\bar{y}_k) \le \phi(y_{k-1})$, then $y_k = \bar{y}_k$. Otherwise $y_k = y_{k-1}$ and the net effect is to have executed (3.2.6ℓ) as is.

Condition (3) may be realized, for example, by using appropriate finite difference operators from §1.2 with $h = \mathcal{O}(\|\phi'(y_0)\|)$ to compute $\bar{f}_k$, π_k, $\omega_j^{(k)}$.

In §3.4 we show one way to construct an approximation $\bar{f}$ to $\phi'(y_0)$ such that $\|\phi'(y_0)\| = \mathcal{O}(\|\bar{f}\|)$ near x^*. With such an $\bar{f}$, (5) implies for any $c \in (0,\infty)$ and $\underline{\mu} \in (0,\mu)$ that

$$\phi(y_k) \le \phi(y_{k-1}) - \frac{1}{2}\frac{\left(\bar{f}_k\right)^2}{\pi_k} + c\|\bar{f}\|^{1+\underline{\mu}}\max\{|\bar{f}_k|, \|\bar{f}\|^{1+\underline{\mu}}\}$$

whenever $\|y_0-x^*\|$ is sufficiently small. This suggests one of the tests incorporated into ψ_1, the modified form of the inner loop of Algorithm (3.2.6) which we now state. While the parameters passed to ψ_1 include y_0, $\|\bar{f}\|$, ξ, and p_1 as well as y_1, all determined by the caller, we will write only $\psi_1(y_1)$ to denote the point y_k returned by ψ_1 $(1 \le k \le n$, depending on the tests in $\psi_1)$.

(8) <u>Algorithm</u> $\psi_1(y_1)$: [inner loop of (3.2.6) plus tests, reordered];

(8a) Let $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $y_0, y_1 \in \mathbb{R}^n$, $\|\bar{f}\| \in (0,\infty)$, $p_1 \in \mathbb{R}^n - \{0\}$, $\xi \in [0,\infty)$, $\mu \in [0,1]$ be given, let the constants $\theta_1 \in (0,1/2]$, and $c_0,\ldots,c_6 \in (0,\infty)$ with $c_5 \in [1,2]$ be fixed, and let δ, ϵ_5, ϵ_6, ϵ_8, ϵ_9 be given;

(8b) $q_1 \leftarrow p_1/\|p_1\|$;

INNERLOOP:

DO k = 2 TO n;

(8c) Choose a basis $u_{k-1}^{(k-1)},\ldots,u_n^{(k-1)}$ for $\mathrm{Span}\{w_1,\ldots,w_{k-2}\}^\perp$ with

(8c(i)) $\chi(U_{k-1}) \le c_0\|U_{k-1}\|$ and

(8c(ii)) $\|u_j^{(k-1)}\| \le c_1$, where

(8c(iii)) $U_{k-1} = \left[u_{k-1}^{(k-1)},\ldots,u_n^{(k-1)}\right]$;

(8d) For $k-1 \le j \le n$ find $\omega_j^{(k-1)}$ such that

$$|\omega_j^{(k-1)} - \frac{\partial^2 \phi}{\partial q_{k-1} \partial u_j^{(k-1)}} (y_{k-2})| \le \epsilon_8 \xi;$$

(8e) $$w_{k-1} \leftarrow U_{k-1}(U_{k-1}^T U_{k-1})^{-1} \begin{pmatrix} \omega_{k-1}^{(k-1)} \\ \vdots \\ \omega_n^{(k-1)} \end{pmatrix};$$

(8f) Choose $p_k \neq 0$ with $\|p_k - P_{k-1}p_k\| \le \delta\xi\|p_k\|$,

where $P_{k-1} = I - W_{k-1}W_{k-1}^+$ and $W_{k-1} = [w_1, \ldots, w_{k-1}]$;

(8g) $\tilde{q}_k \leftarrow (I - Q_{k-1}Q_{k-1}^+)p_k$, where $Q_{k-1} = [q_1, \ldots, q_{k-1}]$;

(8h) $q_k \leftarrow \tilde{q}_k / \|\tilde{q}_k\|$;

(8i) Find $\bar{f}_k$ with $|\bar{f}_k - \frac{\partial \phi}{\partial q_k}(y_{k-1})| \le \epsilon_5 \xi \|\phi'(y_0)\|$;

(8j) Find π_k with $\left|\pi_k - \frac{\partial^2 \phi}{\partial q_k \partial p_k}(y_{k-1})\right| \le \epsilon_6 \xi \|p_k\|$;

(8k) If $\pi_k < c_2\|p_k\|$

(8ℓ) Then Return (y_{k-1});

(8m) Pick $\bar{y}_k \in \bar{B}\left(y_{k-1} - \frac{\bar{f}_k}{\pi_k} p_k, \ \epsilon_9 \xi \|\phi'(y_0)\|^{1+\mu}\right)$;

(8n) If $\|\bar{y}_k - y_{k-1}\| > c_3$ or $\left(\left\| \frac{\bar{f}_k}{\pi_k} p_k \right\| > c_4 \|\bar{f}\|^{c_5}\right.$ and

$$\left. \phi(\bar{y}_k) > \phi(y_{k-1}) - \theta_1 \|p_k\| (\bar{f}_k)^2 / \pi_k + c_6 |\bar{f}_k| \|\bar{f}\|^{c_5} \right)$$

(8o) Then Return (y_{k-1});

(8p) If $\phi(\overline{y}_k) \leq \phi(y_{k-1})$

(8q) Then $y_k \leftarrow \overline{y}_k$;

(8r) Else $y_k \leftarrow y_{k-1}$;

END INNERLOOP;

(8s) Return (y_n);

END ψ_1;

Note that steps (8p,q,r) assure for any $y_1 \in \mathbb{R}^n$ that

(9) $\phi(\psi_1(y_1)) \leq \phi(y_1)$.

As we shall soon see, the y_k given by (8r) satisfies (3.2.6ℓ) near x^* for an appropriate choice of ϵ_7. The calling program constructed in §3.4 reduces to the following form near x^*.

(10) <u>Algorithm</u>: [locally convergent calling program for ψ_1];

(10a) Let $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $\Xi: \mathbb{R}^n \to [0,\infty]$, δ, ϵ_5, ϵ_6, ϵ_8, $\epsilon_9 \in [0,\infty)$, K_1, K_2 with $0 < K_1 < K_2 < \infty$, and $x_0 \in \mathbb{R}^n$ be given and let the constants $\theta_1 \in (0,1/2]$, $c_0, \ldots, c_6 \in (0,\infty)$ with $c_5 \in [1,2]$ be fixed;

OUTERLOOP:

DO $i = 1$ TO ∞;

(10b) If $\phi'(x_{i-1}) = 0$ then STOP and let $x_j = x_{i-1}$ for all $j \geq i$;

(10c) $y_0 \leftarrow x_{i-1}$;

(10d) $\xi \leftarrow \Xi(y_0)$;

(10e) Determine $\bar{f}$ such that $K_1\|\phi'(y_0)\| \leq \|\bar{f}\| \leq K_2\|\phi'(y_0)\|$;

(10f) Choose $p_1 \in \mathbb{R}^n - \{0\}$ and determine $\bar{f}_1, \pi_1$ as in (8i,j) with $k = 1$;

(10g) Pick $y_1 \in B\left(y_0 - \frac{f_1}{\pi_1} p_1, \epsilon_9 \xi \|\phi'(y_0)\|\right)$;

(10h) $x_i \leftarrow \psi_1(y_1)$, with ψ_1 given by (8);

END OUTERLOOP and Algorithm (10);

Note that (10g) requires slightly less accuracy than (8m). The extra accuracy in (8m) is needed because of steps (8p,q,r), as we now see.

(11) Lemma: Suppose $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\phi(x^*) = 0$ with $\phi''(x^*)$ positive definite. Suppose further that there exist $\mu \in (0,1]$, $\rho_2 \in (0,\infty]$, $\kappa \in [0,\infty)$ such that

$$\|\phi''(x) - \phi''(y)\| \leq \kappa\|x-y\|^{\mu} \quad \text{for all} \quad x,y \in B(x^*,\rho_2).$$

Let δ, ϵ_5, ϵ_6, ϵ_8, $\epsilon_9 \in [0,\infty)$, $\theta_1 \in (0,1/2]$, and $c_0,\ldots,c_6 \in (0,\infty)$ with $c_5 \in [1,1+\mu]$ and $c_2 < 1/\|\phi''(x^*)^{-1}\|$ be fixed. Assume $\Xi : \mathbb{R}^n \to [0,\infty]$ satisfies $\Xi(x) = \mathcal{O}(\|\phi'(x)\|^{\mu})$ for all $x \in B(x^*,\rho_2)$. If $c_5 \in [1,1+\mu)$, then there exists $\rho > 0$ such that if $x_0 \in B(x^*,\rho)$, then the sequence $x_0,x_1,\ldots$ generated from x_0 by Algorithm (10) converges to x^* with Q-order $\mathcal{O}_Q(x_0,x_1,\ldots) \geq 1 + \mu$; this convergence rate is strong in the sense of (2.1.66). The same is true if $c_5 = 1+\mu$ and c_6 is sufficiently large.

<u>Proof sketch</u>: By induction on ν we find $\rho^{(\nu)}$, $\epsilon_7^{(\nu)} > 0$ and $c_6^{(\nu)} \geq 0$ such that (8k) and (8n) are false and (3.2.6ℓ) is satisfied with $\epsilon_7 = \epsilon_7^{(\nu)}$ for $1 \leq k \leq \nu$ whenever $y_0 \in B(x^*, \rho^{(\nu)})$, provided $c_6 > c_6^{(\nu)}$. If $c_5 \in [1, 1+\mu)$, then let $c_6^{(\nu)} = 0$, $1 \leq \nu \leq n$. Let $c_6^{(0)} = \epsilon_7^{(0)} = 0$ and $\rho^{(0)} = \rho_2$; suppose $1 \leq \nu \leq n$ and that the above holds with ν replaced by $\nu-1$. Reasoning as in the proof of Lemma (3.2.9) with more care in bounding $|\pi_k|$ and using Lemmas (2.1.20) and (1) we find there exist $\rho^{(\nu)} \in (0, \rho^{(\nu-1)}]$ and $c_6^{(\nu)} \geq c_6^{(\nu-1)}$ such that if $y_0 \in B(x^*, \rho^{(\nu)})$ and $c_6 > c_6^{(\nu)}$ then (8k) and (8n) are false, $1 \leq k \leq \nu$. If (8q) is executed, then (3.2.6ℓ) holds with $\epsilon_7 = \epsilon_9 \max \|\phi'(\bar{B}(x^*, \rho_2))\|^\mu$. Let $\mathcal{y}_k = y_{k-1} - \frac{\bar{f}_k}{\pi_k} p_k$. If (8r) is executed and $|\bar{f}_k| \leq \|\phi'(y_0)\|^{1+\mu}$, then $\|y_k - \mathcal{y}_k\| = \mathcal{O}(\|\phi'(y_0)\|^{1+\mu})$. And if (8r) is executed and $|\bar{f}_k| > \|\phi'(y_0)\|^{1+\mu}$, then Lemma (1) gives $K\|\phi'(y_0)\|^{1+\mu}|\bar{f}_k| > \frac{1}{2}\frac{(\bar{f}_k)^2}{\pi_k}\|p_k\|$, whence $\|y_k - \mathcal{y}_k\| = \|\frac{\bar{f}_k}{\pi_k} p_k\| < 2K\|\phi'(y_0)\|^{1+\mu} = \mathcal{O}(\|\phi'(y_0)\|^{1+\mu})$. Since we may assume WLOG that $\|\phi'(x)\|^\mu = \mathcal{O}(\Xi(x))$ for all $x \in B(x^*, \rho_2)$, we thus see that there exists $\epsilon_7^{(\nu)} \geq \epsilon_7^{(\nu-1)}$ such that (3.2.6ℓ) is satisfied with $\epsilon_7 = \epsilon_7^{(\nu)}$, $1 \leq k \leq \nu$. This completes the induction.

With Ξ modified in such a way that $\Xi(x) = +\infty$ whenever $\|x - x^*\| \geq \rho^{(n)}$, we see that Algorithm (10) reduces to Algorithm (3.2.6) with $\epsilon_7 = \epsilon_7^{(n)}$, provided $c_6 > c_6^{(v)}$. Thus the conclusions of (11) follow from Lemma (3.2.9). ∎

It is hoped when $c_2, \ldots, c_6$ are properly chosen that steps (8ℓ) and (8o) will tend to be executed if and only if the effort required to complete the remaining minor iterations in ψ_1 would not be worthwhile. Test (8k) also ensures that $\pi_k > 0$ and thus that steps (8m,n) are well defined. The first test in (8n) was included to prevent "unreasonably" large steps. Regarding the last two tests in (8n), it should happen only rarely that $\left\| \frac{\bar{f}_k}{\pi_k} p_k \right\| \leq c_4 \|\bar{f}\|^{c_5}$, in which case the last test in (8n) might fail for arbitrarily small values of $\|y_0 - x^*\|$. But in this case $|\phi(\bar{y}_k) - \phi(y_{k-1})|$ should be very small, so we choose to pass directly to step (8p). For small $\|y_0 - x^*\|$, test (8p) should be necessary only when $|\bar{f}_k| / \|\bar{f}\|$ is small, in which case truncation and roundoff error might force (8r) to be executed.

The best choices (if they exist) for $c_2, \ldots, c_6$ depend on ϕ and x^* and must in general be determined experimentally. Preliminary computational experience is too meager to say whether choices for these constants exist which perform well on a broad range of $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$.

For completeness we state without proof what Lemma (11) becomes in the case $\mu = 0$.

(12) Lemma: Suppose $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\phi'(x^*) = 0$ with $\phi''(x^*)$ positive definite. Let $\delta, \epsilon_5, \epsilon_6, \epsilon_8, \epsilon_9 \in [0,\infty)$, $\theta_1 \in (0,1/2]$, and $c_0, \ldots, c_6 \in (0,\infty)$ with $c_5 = 1$ and $c_2 < 1/\|\phi''(x^*)\|$ be fixed. There

exist $\rho_0, \xi_1 > 0$ such that if $\rho \in (0,\rho_0]$, $\Xi: \mathbb{R}^n \to [0,\infty]$ satisfies $\sup \Xi(B(x^*,\rho)) \le \xi_1$, and $x_0 \in B(x^*,\rho)$, then the sequence $x_0, x_1, \ldots$ of major iterates generated from x_0 by Algorithm (10) converges (at least) Q-linearly to x^*. Moreover, if $\lim_{h\to 0} [\sup \Xi(B(x^*,h))] = 0$, then the convergence is Q-superlinear. ∎

3.4 A Globally Convergent Calling Program

In this subsection we construct an algorithm which, under reasonable conditions, produces a sequence $x_0, x_1, \ldots$ that converges to a critical point x^* of $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$. If x^* is a nonsingular minimizer and certain control constants lie within appropriate ranges of values, then the algorithm calls ψ_1, i.e. Algorithm (3.3.8), near x^* and the convergence to x^* is rapid in the sense of Lemmas (3.3.11) or (3.3.12). Because the calling program requires an approximation $\bar{f} = \tilde{f}(x,h)$ to $\phi'(x)$ which satisfies certain conditions, we first show one way to construct such an approximation.

Forward differences suffice for constructing an approximation to $\phi'(x)$ having the desired properties. Given $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, let

(1) $$\tilde{f}(x,h) = \left(\frac{\Delta\phi}{\Delta e_1}(x,h), \ldots, \frac{\Delta\phi}{\Delta e_n}(x,h)\right)^T,$$

where $\frac{\Delta\phi}{\Delta u}$ is defined in (1.2.1). The desired properties are described in

(2) Lemma: Assume $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and let $\tilde{f}: \mathbb{R}^n \times (0,\infty) \to \mathbb{R}^n$ be given by (1). For each $x \in \mathbb{R}^n$ and $\zeta \in (0,\infty)$ there exist $\rho_1, h_1 \in (0,\infty)$

such that

(3) $\|\nabla\phi(x) - \tilde{f}(y,h)\| < \zeta$ whenever $y \in B(x,\rho_1)$ and $h \in (0,h_1]$.

Moreover, if $\underline{h}, h, K, \gamma_0 \in (0,\infty)$ and $x \in \mathbb{R}^n$ are such that $\phi'(x) \neq 0$,

(4a) $\underline{h} \leq h \leq \gamma_0 \|\tilde{f}(x,h)\|$,

(4b) $\max\|\phi''(\bar{B}(x,h))\| \leq K$, and

(4c) $nK\gamma_0 < 1$,

then

(5) $$\frac{1-2nK\gamma_0}{1-nK\gamma_0}\|\phi'(x)\| \leq \|\tilde{f}(x,\underline{h})\| \leq \frac{\|\phi'(x)\|}{1-nK\gamma_0}$$ and

(6) $$\|\nabla\phi(x) - \tilde{f}(x,\underline{h})\| \leq \left(\frac{nK\gamma_0}{1-nK\gamma_0}\right)\|\phi'(x)\|.$$

<u>Proof</u>: Let $x \in \mathbb{R}^n$ and $\zeta \in (0,\infty)$ be given. By the continuity of ϕ' there exist $\rho_1, h_1 > 0$ such that $\|\phi'(z) - \phi'(x)\| \leq \zeta/\sqrt{n}$ for all $z \in B(x,\rho_1+h_1)$. Since $\frac{\Delta\phi}{\Delta e_i}(y,h) = \frac{\phi(y+he_i)-\phi(y)}{h}$, we may use the MVT to show that (3) holds.

Assume now that (4) holds with $\underline{h} > 0$. As in the proof of Lemma (3.2.2), we find

(7) $\|\tilde{f}(x,\underline{h}) - \nabla\phi(x)\| \leq nK\underline{h}$.

In particular, $\|\tilde{f}(x,h) - \nabla\phi(x)\| \leq nKh \leq nK\gamma_0\|\tilde{f}(x,h)\|$ by (4a), so $\|\phi'(x)\| \geq \|\tilde{f}(x,h)\|(1-nK\gamma_0)$, which with (4c) gives

(8) $$\|\tilde{f}(x,h)\| \leq \frac{\|\phi'(x)\|}{1-nK\gamma_0} .$$

From (4a), (7), and (8) we have

$$\|\tilde{f}(x,\underline{h})\| \leq \|\phi'(x)\| + nK\underline{h} \leq \|\phi'(x)\| + nK\gamma_0\|\tilde{f}(x,h)\|$$
$$\leq \|\phi'(x)\| \left[1 + \frac{nK\gamma_0}{1-nK\gamma_0}\right],$$

whence follows the second inequality in (5). From (4a), (7), and (8) we also obtain

$$\|\tilde{f}(x,\underline{h})\| \geq \|\phi'(x)\| - nK\underline{h} \geq \|\phi'(x)\| - nK\gamma_0\|\tilde{f}(x,h)\|$$
$$\geq \|\phi'(x)\| \left[1 - \frac{nK\gamma_0}{1-nK\gamma_0}\right],$$

which implies the first inequality in (5). Finally, (4a), (7), and (8) imply $\|\nabla\phi(x)-\tilde{f}(x,\underline{h})\| \leq nK\gamma_0\|\tilde{f}(x,h)\| \leq \left(\frac{nK\gamma_0}{1-nK\gamma_0}\right)\|\phi'(x)\|$, which is just (6). ∎

Note that if $\underline{h} = h$, then we see from (7) and (4a) that we can replace the first inequality in (5) by the stronger one $\frac{\|\phi'(x)\|}{1+nK\gamma_0} \leq \|\tilde{f}(x,h)\|$.

Property (3) enables us to devise a descent algorithm (e.g. (9)) which produces a sequence of iterates having no noncritical point as a point of accumulation.

Given $\emptyset \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x_0 \in \mathbb{R}^n$, it frequently happens that the level set $L(x_0) = \{x \in \mathbb{R}^n \mid \emptyset(x) \le \emptyset(x_0)\}$ may be expressed as $C_1 \cup C_2$ with $x_0 \in C_1$, $\overline{c}_3 \equiv \inf\{\|x-y\|: x \in C_1, y \in C_2\} > 0$, and $K \equiv \sup\{\|\emptyset''(x)\|: \|x-C_1\| \le h_0\} < +\infty$ for some $h_0 > 0$, where $\|x-C_1\| \equiv \inf\{\|x-y\|: y \in C_1\}$. In this case any descent method which takes steps of length $< \overline{c}_3$ will produce only iterates in C_1 when started at x_0. And if $x_{i-1} \in C_1$, $h \le h_0$, $nK\gamma_0 < 1/2$, and (4a) holds, then (3.3.10e) is satisfied with $\bar{f} = \tilde{f}(x_{i-1}, \underline{h})$, $K_1 = \dfrac{1-2nK\gamma_0}{1-nK\gamma_0}$, and $K_2 = 1/(1-nK\gamma_0)$.

Thus under fairly reasonable conditions condition (3.3.10e) may be met by choosing sufficiently small $h_0, \gamma_0 > 0$, trying successively smaller values of $h > 0$ until $h \le \gamma_0\|\tilde{f}(x_{i-1}, h)\|$, and letting $\bar{f} = \tilde{f}(x_{i-1}, \underline{h})$ for any $\underline{h} \in (0,h]$. Moreover, under these reasonable conditions Lemma (2) implies that if $\gamma_0 > 0$ is sufficiently small, then the $\tilde{f}$ given by (1) satisfies requirements (9a(iv-viii)) below with $\zeta_1 = \dfrac{nK\gamma_0}{1-nK\gamma_0}$.

Recall that the ψ_1 given by Algorithm (3.3.8) has $\emptyset$ among its implicit parameters and thus has no trouble satisfying (3.3.9), i.e. $\emptyset(\psi_1(y_1)) \le \emptyset(y_1)$ for any $y_1 \in \mathbb{R}^n$. In Algorithm (9), the calling program for ψ_1 to which we now turn, we need only assume that ψ_1 is any subroutine with this property.

(9) <u>Algorithm</u>: [Globally convergent calling program for ψ_1];

(9a) Let $\emptyset \in C^2(\mathbb{R}^n, \mathbb{R})$, $\tilde{f}: \mathbb{R}^n \times (0,\infty) \to \mathbb{R}^n$, $\Xi: \mathbb{R}^n \to [0,\infty]$,

$\psi_1: \mathbb{R}^n \to \mathbb{R}^n$, $K_1, \epsilon_5, \epsilon_6, \epsilon_8, \epsilon_9, c_2, \ldots, c_7, h_0, \gamma_0 \in (0,\infty)$, $\theta_1 \in (0,1/2]$, $\zeta_1, \theta_2, \ldots, \theta_5 \in (0,1)$, $\ell_1, \ldots, \ell_4 \in \mathbb{N}$, and $x_0 \in \mathbb{R}^n$ be given such that the following conditions hold:

(9a(i)) $\phi(\psi_1(y_1)) \leq \phi(y_1)$,

(9a(ii)) $c_6 < \theta_1 c_4$,

(9a(iii)) $\ell_2 \leq \ell_4 \geq 1$ and $\ell_1 \leq \ell_3$,

and, setting $\hat{L}(x_0) = \left\{ x \in \mathbb{R}^n \mid \phi(x) \leq \phi(x_0) \text{ and } \phi'(x) \neq 0 \right\}$,

(9a(iv)) for each $x \in \hat{L}(x_0)$ there exists $h > 0$ with $h \leq \gamma_0 \|\tilde{f}(x,h)\|$,

(9a(v)) $K_1 \|\phi'(x)\| \leq \|\tilde{f}(x,\underline{h})\|$ whenever $x \in \hat{L}(x_0)$ and $0 < \underline{h} \leq h \leq \gamma_0 \|\tilde{f}(x,h)\|$,

(9a(vi)) $\zeta_1 < \dfrac{\theta_2}{1+\theta_2}$,

(9a(vii)) $\|\nabla\phi(x) - \tilde{f}(x,\underline{h})\| \leq \zeta_1 \|\phi'(x)\|$ whenever $x \in \hat{L}(x_0)$ and $0 < \underline{h} \leq h \leq \gamma_0 \|\tilde{f}(x,h)\|$,

(9a(viii)) and for any $x \in \hat{L}(x_0)$ and $\zeta > 0$ there exist $h_1, \rho_1 > 0$ such that (3) holds;

OUTERLOOP:

DO $i = 1$ TO ∞;

(9b) If $\phi'(y_0) = 0$ then stop and let $x_j = x_{i-1}$ for all $j \geq i$;

(9c) $y_0 \leftarrow x_{i-1}$;

(9d) $\xi \leftarrow \Xi(y_0)$;

(9e) $\ell \leftarrow 0$;

(9f) Find $h^{(0)} \in (0,h_0]$ such that $h^{(0)} \leq \gamma_0\|\tilde{f}(y_0,h^{(0)})\|$;

REPEAT:

(9g) $\bar{f} \leftarrow f(y_0,h^{(\ )})$;

(9h) Choose $p_1 \in \mathbb{R}^n$ with

(9h(i)) $\|p_1\| \geq c_7\|\bar{f}\|$ and

(9h(ii)) $|p_1^T\bar{f}| \geq \theta_2\|p_1\|\|\bar{f}\|$;

(9i) $q_1 \leftarrow p_1/\|p_1\|$;

(9j) Find π_1 with $|\pi_1 - \frac{\partial^2\phi}{\partial q_1 \partial p_1}(y_0)| \leq \epsilon_6\xi\|p_1\|$;

(9k) If $\pi_1 < c_2\|p_1\|$ then GO TO ARMIJO;

(9ℓ) Find $\bar{f}_1$ with $|\bar{f}_1 - \frac{\partial\phi}{\partial q_1}(y_0)| \leq \epsilon_5\xi\|\phi'(y_0)\|$;

(9m) Pick $y_1 \in B(y_0 - \frac{\bar{f}_1}{\pi_1}p_1,\ \epsilon_9\xi\|\phi'(y_0)\|)$;

(9n) If $\|y_1-y_0\| > c_3$ or $\left\|\frac{\bar{f}_1}{\pi_1}\, p_1\right\| \leq c_4\|\bar{f}\|^{c_5}$ or

$$\phi(y_1) > \phi(y_0) - \theta_1\|p_1\|(\bar{f}_1)^2/\pi_1 + c_6|\bar{f}_1|\|\bar{f}\|^{c_5}$$

(9o) Then GO TO ARMIJO;

(9p) $x_i \leftarrow \psi_1(y_1)$;

(9q) GO TO LOOPEND;

ARMIJO:

DO $j = \ell_1 + (\ell_2)\ell$ TO $\ell_3 + (\ell_4)\ell$;

(9r) $t_{\ell,j} \leftarrow \theta_3^j \operatorname{sgn}(\bar{f}^T p_1)$, where $\operatorname{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}$;

(9s) $x_i \leftarrow x_{i-1} - t_{\ell,j} p_1$;

(9t) If $\|x_i - x_{i-1}\| \leq c_3$ and

$\phi(x_i) \leq \phi(x_{i-1}) - t_{\ell,j}\theta_4(\bar{f}^T p_1)$

(9u) Then GO TO LOOPEND;

END ARMIJO;

(9v) $\ell \leftarrow \ell+1$;

(9w) $h^{(\ell)} \leftarrow \theta_5 h^{(\ell-1)}$;

(9x) GO TO REPEAT;

LOOPEND:

END OUTERLOOP and Algorithm (9);

When x_i is not determined by ψ_1 in step (9p), the technique used to determine x_i is a slight generalization of one proposed by Armijo (1966). If $\tilde{f}$ is well behaved, as is the $\tilde{f}$ with which Lemma (2) deals, then an acceptable x_i will eventually be found whenever $\phi'(x_{i-1}) \neq 0$. Step (9b) ensures that Algorithm (9) produces an infinite sequence even if one of the major iterates should chance to be a critical point of ϕ.

Algorithm (9) is globally convergent in the following sense. (Note that (9a(vi,vii)) and the inequality $\theta_1 \leq 1/2$ are not needed for Lemma (10), but Theorem (24) does rely upon them.)

(10) Lemma: If (9a) holds, then Algorithm (9) produces an infinite sequence $x_0, x_1, \ldots$ such that $\phi(x_i) \leq \phi(x_{i-1})$ for all $i \geq 1$. Moreover, if x^* is an accumulation point of this sequence, then $\phi'(x^*) = 0$.

Proof: First we show that x_i, once found, satisfies $\phi(x_i) \leq \phi(x_{i-1})$. If (9p) is executed, then (9k) implies $\pi_1 > 0$ and the second test in (9n) implies $\|p_1\|(\bar{f}_1)^2/\pi_1 > c_4|\bar{f}_1|\|\bar{f}\|^{c_5} > 0$, so the third test in (9n) combines with (9a(ii)) to yield

$$
\begin{aligned}
\phi(y_1) &\leq \phi(y_0) - \theta_1\|p_1\|(\bar{f}_1)^2/\pi_1 + c_6|\bar{f}_1|\|\bar{f}\|^{c_5} \\
&\leq \phi(y_0) - (\theta_1 c_4 - c_6)|\bar{f}_1|\|\bar{f}\|^{c_5} \qquad (15) \\
&< \phi(y_0).
\end{aligned}
$$

Because of (9a(i)) we thus have $\phi(x_i) < \phi(x_{i-1})$ if (9p) is executed. On the other hand, if (9u) is executed, then (9r), (9t), and (9h(ii)) imply $\phi(x_i) < \phi(x_{i-1})$. Thus $\phi(x_i) < \phi(x_{i-1})$ unless $\phi'(x_{i-1}) = 0$, in which case $\phi(x_i) = \phi(x_{i-1})$ by step (9b).

In what follows we will see that if $\phi'(x_{i-1}) \neq 0$ and (9p) is never executed, then (9u) must eventually be executed during the i^{th} major iteration. Thus (9) does in fact generate an infinite sequence.

To complete the proof, it suffices to show that if $\phi'(z) \neq 0$, then there exists $\rho > 0$ such that

(16) $$\phi(x_i) < \min \phi(\bar{B}(z,\rho)) \quad \text{whenever} \quad x_{i-1} \in B(z,\rho).$$

Fix $z \in \mathbb{R}^n$ with $\phi'(z) \neq 0$. Let $\rho_1 > 0$ be small enough that

(17) $$\min\|\phi'(\bar{B}(z,\rho_1))\| \geq \frac{1}{2}\|\phi'(z)\|.$$

Consider first what happens if (9p) is executed with $y_0 = x_{i-1} \in B(z,\rho_1)$. Step (9k) implies $\pi_1/\|p_1\| \geq c_2$; from the second test of (9n) we therefore have $|\bar{f}_1| > \frac{\pi_1}{\|p_1\|} c_4\|\bar{f}\|^{c_5} \geq c_2c_4\|\bar{f}\|^{c_5}$. Together with (9a(v)) and (9g), this implies $|\bar{f}_1|\|\bar{f}\|^{c_5} > c_2c_4\|\bar{f}\|^{2c_5} \geq c_2c_4K_1^{2c_5}\|\phi'(y_0)\|^{2c_5}$. Putting this into (15) and using (17) and (9a(i)) we thus find

(18) $$\phi(x_i) \leq \phi(x_{i-1}) - (\theta_1c_4-c_6)c_2c_4\|\tfrac{1}{2}K_1\phi'(z)\|^{2c_5}$$

whenever (9p) is executed with $x_{i-1} \in B(z,\rho_1)$.

Consider now what occurs in the ARMIJO loop when $y_0 = x_{i-1} \in B(z,\rho_1)$. WLOG $p_1^T\bar{f} > 0$. (Note that (9a(v)) and (9h(ii)) imply $p_1^T\bar{f} \neq 0$.) Using (9h(ii)) and (9a(v)) we find

$$p_1^T\bar{f} \geq \theta_2\|p_1\|\|\bar{f}\| \geq K_1\theta_2\|p_1\|\|\phi(y_0)\|.$$

Combined with (17), this gives

(19) $$p_1^T\bar{f} \geq K_1\theta_2\|p_1\|\|\phi'(z)\|/2$$

whenever $y_0 = x_{i-1} \in B(z,\rho_1)$.

Assume $\|\nabla\phi(y) - \bar{f}\| \leq \zeta$. Using (9h(ii)) and (9a(v)), we find:

$$(20)\qquad \begin{aligned} p_1^T\nabla\phi(y) &= p_1^T\bar{f} + p_1^T(\nabla\phi(y) - \bar{f}) \\ &\geq p_1^T\bar{f} - \|p_1\|\zeta \\ &\geq p_1^T\bar{f}\left(1 - \frac{\zeta}{\theta_2\|\bar{f}\|}\right) \\ &\geq p_1^T\bar{f}\left(1 - \frac{\zeta}{K_1\theta_2\|\phi'(y_0)\|}\right). \end{aligned}$$

By (9a(v)) we may assume ρ_1 to be so small that for some $h_1 > 0$ (3) holds with $x = z$ and $\zeta = (1-\theta_4)K_1\theta_2\|\phi'(z)\|/2$. Together with (17) and (20), this yields

$$(21)\qquad \phi'(y)p_1 \geq \theta_4 p_1^T\bar{f}$$

whenever $y \in B(z,\rho_1)$, $y_0 = x_{i-1} \in B(z,\rho_1)$, and $h^{(\ell)} \leq h_1$.

Steps (9w) and (9f) imply $h^{(\ell)} \leq \theta_5^\ell h_0$. Thus there exists $\ell^* \in \mathbb{N}$ such that $h^{(\ell)} \leq h_1$ for all $\ell \geq \ell^*$. For any $x_{i-1} = y_0 \in B(z,\rho_1/2)$, let $\Lambda(y_0)$ denote the least $\ell \geq \ell^*$ such that $\theta_3^{\ell_3+(\ell_4)\ell}\|p_1\| \leq \rho_1/2$: by (9a(iii)) the range of the ARMIJO loop includes $j = \ell_3 + (\ell_4)\ell$, which tends to $+\infty$ as $\ell \to +\infty$, so $\Lambda(y_0)$ exists. From (9h(i)), (9a(v)), and (17) we see that

$$(22)\qquad \theta_3^{\ell_3+(\ell_4)\Lambda(y_0)}\|p_1\| \geq \min\left\{\theta_3^{\ell_4}\rho_1/2, \left[c_7K_1\|\phi'(z)\|\theta_3^{\ell_3+\ell_4\ell^*}\right]/2\right\} \equiv K_3.$$

WLOG $\rho_1/2 \leq c_3$. Thus if $y_0 = x_{i-1} \in B(z,\rho_1/2)$ and (9s) is executed with $\ell = \Lambda(y_0)$ and $j = \ell_3 + (\ell_4)\ell$, then $y = x_{i-1} + \lambda(x_i - x_{i-1}) \in B(z,\rho_1)$ for all $\lambda \in [0,1]$, so (21) implies that (9t) is true. Hence from (19), (9r), (9t), and (22) we see that

$$(23) \qquad \emptyset(x_i) \leq \emptyset(x_{i-1}) - K_1\theta_2K_3\theta_4\|\emptyset'(z)\|/2$$

whenever $x_{i-1} \in B(z,\rho_1/2)$ and (9p) is not executed during the i^{th} major iteration.

From (18) and (23) we see that there exists $d > 0$ such that $\emptyset(x_i) \leq \emptyset(x_{i-1}) - d$ whenever $x_{i-1} \in B(z,\rho_1/2)$. By the continuity of $\emptyset$ there exists $\rho \in (0,\rho_1/2]$ such that

$$\max \emptyset(\bar{B}(z,\rho)) - \min \emptyset(\bar{B}(z,\rho)) < d.$$

Thus if $x_{i-1} \in B(z,\rho)$, then

$$\emptyset(x_i) \leq \emptyset(x_{i-1}) - d \leq \max \emptyset(\bar{B}(z,\rho)) - d$$
$$< \min \emptyset(\bar{B}(z,\rho)),$$

which is just (16). ∎

Lemma (10) implies in particular that if x_0 lies in a compact component C_1 of the level set $L(x_0) = \{x \in \mathbb{R}^n \mid \emptyset(x) \leq \emptyset(x_0)\}$ and if c_3 is small enough and Ψ_1 is such that all major iterates remain in C_1, then $\lim_{i\to\infty} \emptyset'(x_i) = 0$. Because (9) is a descent algorithm, if x^* is both a nonsingular minimizer of $\emptyset$ and an accumulation point of the sequence $x_0, x_1, \ldots$, and if there exists $K_2 \in (0,\infty)$ such that $\|p_1\| \leq K_2\|\emptyset'(x_{i-1})\|$ in the i^{th} major iteration, then $x^* = \lim_{i\to\infty} x_i$.

In step (9s) we assumed exact arithmetic. However, similar reasoning shows that if $x_i \in B\left(x_{i-1} - t_{\ell,j}p_1 ; \Lambda(\ell,h^{(\ell)})\right)$, where $\Lambda(\ell,h^{(\ell)})/\min\{h^{(\ell)},h^{(0)}\theta_3^{\ell_3+(\ell_4)\ell}\} \to 0$ as $\min\{h^{(\ell)},h^{(0)}\theta_3^{\ell_3+(\ell_4)\ell}\} \to 0$, then

Lemma (10) still holds.

We have seen that under reasonable circumstances $\tilde{f}$ may be calculated by forward differences, as in (1). Note that the $\bar{f}_1$ found in (9ℓ) must be somewhat more accurate than $p_1^T \bar{f}$; for example, $\bar{f}_i$ may be computed by central differences.

We come now to the main result of this subsection, which shows under reasonable conditions that Algorithm (9) is globally Q-quadratically convergent when Algorithm (3.3.8) is used for ψ_i. Note that ζ_1, c_2, c_4, c_5, c_6, and θ_2 are among the constants specified in (9a).

(24) <u>Theorem</u>: Let Algorithm (9) run with ψ_i given by Algorithm (3.3.8). If $x_0 \in \mathbb{R}^n$ is such that (9a) holds, then the sequence $x_0, x_1, \ldots$ of major iterates produced from x_0 is such that $\phi(x_i) \le \phi(x_{i-1})$ for all $i \ge 1$, and any accumulation point x^* of this sequence satisfies $\phi'(x^*) = 0$. Moreover, if x^* is such an accumulation point with $\phi''(x^*)$ positive definite and $\|\phi''(x^*)^{-1}\| < 1/c_2$, if there exist $\mu \in (0,1]$, $\rho_2 \in (0,\infty]$, and $\kappa \in [0,\infty)$ such that

(25a) $$\|\phi''(x) - \phi''(y)\| \le \kappa \|x-y\|^\mu \text{ for all } x,y \in B(x^*,\rho_2) \text{ and}$$

(25b) $$\Xi(x) = \mathcal{O}(\|\phi'(x)\|^\mu) \text{ for all } x \in B(x^*,\rho_2),$$

and if either

(27a) $$c_5 = 1 + \mu \text{ and } c_6 \text{ is sufficiently large or}$$

(27b) $$c_5 \in (1, 1+\mu) \text{ or}$$

(27c) $$c_5 = 1 \text{ and } \theta_2 - \frac{1}{1-\zeta_1} > c_4 \|\phi''(x^*)\|,$$

then the major iterates $x_0, x_1, \ldots$ converge to x^* with Q-order $O_Q(x_0, x_1, \ldots) \geq 1 + \mu$. This convergence rate is strong in the sense of (2.1 66).

<u>Proof</u>: Assume $\phi'(x^*) = 0$ with $\phi''(x^*)$ positive definite. Note that the second inequality in (3.3.10e) holds with $K_2 = 1 + \zeta_1$, by virtue of (9f), (9g), and (9a(vii)). Because of Lemmas (10) and (3.3.11) and fact (3.3.9), it suffices to show that x^* has a neighborhood $B(x^*,\rho)$ (with $\rho > 0$) such that (9p) is executed whenever $y_0 = x_{i-1} \in B(x^*,\rho)$. To start with, (25b), (9d), (9j), and the hypothesis $c_2 < 1/\|\phi''(x^*)^{-1}\|$ imply the existence of $\rho > 0$ such that $\pi_1 \geq c_2\|p_1\|$ if $y_0 \in B(x^*,\rho)$, whence (9k) is false for such y_0. From (9ℓ), (9m), (9d), and (25b) we may assume ρ to be so small that $\|y_1 - y_0\| \leq c_3$ (in step (9n)) whenever $y_0 \in B(x^*,\rho)$. In view of Lemma (3.3.1), it suffices to show that if ρ is small enough then

$$\text{(28)} \qquad \left\| \frac{\bar{f}_1}{\pi_1} p_1 \right\| > c_4 \|\bar{f}\|^{c_5} \quad \text{whenever } y_0 \in B(x^*,\rho),\ y_0 \neq x^*.$$

WLOG $\frac{\partial\phi}{\partial q_1}(y_0) = \phi(y_0)p_1/\|p_1\| \geq 0$. Note that (9f), (9g), and (9a(vii)) imply $\|\phi'(y_0)\| \leq \|\bar{f}\| + \zeta_1\|\phi'(y_0)\|$, whence (since $\zeta_1 < 1$ by (9a(vi))) $\|\phi'(y_0)\| \leq \|\bar{f}\|/(1-\zeta_1)$. Together with (9ℓ), (9h(ii)), and (9a(vii)), this yields

$$\bar{f}_1 \geq \frac{\phi'(y_0)p_1}{\|p_1\|} - \epsilon_5\xi\|\phi'(y_0)\| \geq \frac{p_1^T\bar{f}}{\|p_1\|} - \|\nabla\phi(y_0) - \bar{f}\| - \epsilon_5\xi\|\phi'(y_0)\|.$$

$$\geq \theta_2\|\bar{f}\| - (\zeta_1 + \epsilon_5\xi)\|\phi'(y_0)\| \geq \|\bar{f}\|\left(\theta_2 - \frac{\zeta_1+\epsilon_5\xi}{1-\zeta_1}\right).$$

Since $\theta_2 > \frac{\zeta_1}{1-\zeta_1}$ by (9a(vi)), (25b) et al. imply that we may assume ρ to be small enough that

$$\bar{f}_1 \geq \|\bar{f}\|\left(\theta_2 - \frac{\zeta_1+\epsilon_5\xi}{1-\zeta_1}\right) > 0 \quad \text{if} \quad y_0 \in B(x^*,\rho).$$

Now for any $\epsilon > 0$ we see from (9j) that $\rho > 0$ may be chosen so small that $\left\|\frac{p_1}{\pi_1}\right\| \geq \frac{1-\epsilon}{\|\phi''(x^*)\|}$ whenever $y_0 \in B(x^*,\rho)$. For such y_0 we have

$$\left\|\frac{\bar{f}_1}{\pi_1} p_1\right\| \geq \|\bar{f}\|\left(\theta_2 - \frac{\zeta_1+\epsilon_5\xi}{1-\zeta}\right)\frac{(1-\epsilon)}{\|\phi''(x^*)\|} .$$

Hence if (27) holds, then by (25b) we see that (28) holds for any sufficiently small $\rho > 0$. ∎

For completeness, we state a result for the case $\mu = 0$. The omitted proof is similar to the one above but relies in part on Lemma (3.3.12) rather than (3.3.11).

(29) Theorem: Assume the hypotheses of Theorem (24) are valid, except for those relating to μ, such as (25). Let $c_5 = 1$ and suppose $\theta_2 - \frac{\zeta_1}{1-\zeta_1} > c_4\|\phi''(x^*)\|$. There exists $\xi_1 > 0$ such that if $\Xi(x) \leq \xi_1$ for all $x \in B(x^*,\rho)$ for some $\rho > 0$, then the major iterates $x_0, x_1, \ldots$ converge at least Q-linearly to x^*. Moreover, if $\lim_{h\to 0}[\sup \Xi(B(x^*,h))] = 0$,

then $x_0, x_1, \ldots$ converge Q-superlinearly to x^*. ∎

3.5 Specific Choices for p_k and $u_j^{(k)}$

In this subsection we suggest specific ways to choose p_k in Algorithms (3.3.8) and (3.4.9) and to generate $u_k^{(k)}, \ldots, u_n^{(k)}$ in Algorithm (3.3.8). One of the methods we thus propose may be viewed as a nonlinear Cholesky decomposition algorithm and resembles Brown's method in that it maintains the basis $u_k^{(k)}, \ldots, u_n^{(k)}$ in the same way and chooses $p_k = u_k^{(k)}$. (It differs from Brown's method in the choice of q_k.) Another one is very similar to various nonlinear conjugate gradient algorithms; we explore the relationship between these methods and see thereby how to implement finite difference forms of the conjugate gradient algorithms. We conclude by commenting briefly on the arithmetic accuracy needed for various calculations when finite but variable precision arithmetic is used.

First we consider the following nonlinear Cholesky decomposition method:

(1) Algorithm: [nonlinear Cholesky decomposition];

(1a) Let the quantities described in (3.4.9a) and (3.3.8a) be given and run Algorithm (3.4.9), using (3.3.8) for ψ_1 and making the following choices;

(1b) In (3.4.9h) reorder the variables if necessary so that $|e_1^T\bar{f}| = \|\bar{f}\|_\infty$ and let $p_1 = (e_1^T\bar{f})e_1$;

(1c) To carry out (3.3.8c), let $u_j^{(1)} = e_j$, $1 \le j \le n$, and in place of (3.3.8e) compute

$$u_j^{(k)} = u_j^{(k-1)} - \left(\frac{\omega_j^{(k-1)}}{\omega_{k-1}^{(k-1)}}\right) u_{k-1}^{(k-1)}, \qquad k \le j \le n;$$

(1d) Choose $p_k = u_k^{(k)}$ in step (3.3.8f).

END Algorithm (1);

Algorithm (1) generates $u_j^{(k)}$ in the same way as Brown's method when (2.2.12i) is used. As with Brown's method, it would also be possible to generate $u_j^{(k)}$ by backsolving $\Omega_k u_j^{(k)} = e_j$, where Ω_k is given by (2.2.9). However, this would increase the storage required for producing the $u_j^{(k)}$ from $n^2/4 + \mathcal{O}(n)$ to $n^2/2 + \mathcal{O}(n)$ and would increase the number of arithmetic operations per complete execution of Ψ_1 from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^4)$.

The choice of p_1 made in (1b) satisfies conditions (3.4.9h(i,ii)) with $c_7 = \theta_2 = 1/\sqrt{n}$. When exact arithmetic is used, the choice of p_k, $2 \le k \le n$, made in (1b) satisfies (3.3.8f) with $\delta = 0$. Since $U_k = [u_k^{(k)}, \ldots, u_n^{(k)}]$ has the form

(2) $$U_k = \begin{bmatrix} \bar{U}_k \\ \cdots \\ I_{n-k+1} \end{bmatrix}; \quad \text{where } \bar{U}_k \in \mathcal{L}(\mathbb{R}^{n-k+1}, \mathbb{R}^{k-1}),$$

the smallest singular value $\|U_k\|/\chi(U_k)$ of U_k is not less than 1, so (3.3.8c(i)) holds with $c_0 = 1$. After examining the behavior of Algorithm (1) when ϕ is quadratic and $\xi = 0$, we will see that if x^* is a nonsingular minimizer of ϕ, then x^* has a neighborhood $B(x^*, \rho)$ such that

(3.3.8c(ii)) holds whenever $y_0 \in B(x^*,\rho)$. Thus the choices of p_k and $u_j^{(k)}$ made in (1) meet the requirements of (3.3.8) and (3.4.9) (with $\Xi(x) = +\infty$ for $\|x-x^*\| > \rho$), so Theorem (3.4.24) applies.

Let us now see how Algorithm (1) behaves if $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ is quadratic with $\phi''(x) \equiv A$ positive definite and if exact calculations are performed, i.e. $\Xi \equiv 0$. Assume no reordering is done in (1b) and let

(3) $$\mathcal{U}^{(k)} = [u_1^{(1)}, u_2^{(2)}, \ldots, u_k^{(k)}, u_{k+1}^{(k)}, \ldots, u_n^{(k)}],$$

as in (2.2.13). Note that $\dfrac{\partial^2\phi}{\partial p_k \partial u_j^{(k)}} = u_k^{(k)T} A u_j^{(k)} = \|\tilde{q}_k\| \omega_j^{(k)}$, where $\tilde{q}_k$

is given by (3.3.8g)--see e.g. (3.2.23), which assumes p_k to be scaled in such a way that $\|\tilde{q}_k\| = 1$. From (1c) we see that (2.2.16) holds, and by induction on k we find that

(4a) $$\mathcal{U}^{(k)T} A \mathcal{U}^{(k)} = D^{(k)},$$

where

(4b) $$D^{(k)} = \begin{bmatrix} d_1 & 0 & \cdots & & & \cdots & 0 \\ 0 & d_2 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & & & & \vdots \\ & & & d_{k-1} & 0 & 0 \cdots & 0 \\ & & \ddots & 0 & \tilde{\omega}_k^{(k)} & \tilde{\omega}_{k+1}^{(k)} \cdots & \tilde{\omega}_n^{(k)} \\ \vdots & & & 0 & \tilde{\omega}_{k+1}^{(k)} & X \cdots & X \\ \vdots & & & \vdots & \vdots & \vdots \ddots & \vdots \\ 0 & \cdots & & 0 & \tilde{\omega}_n^{(k)} & X \cdots & X \end{bmatrix},$$

(4c) $\quad d_k = \hat{d}_k^{(k)},$

(4d) $\quad \tilde{\omega}_j^{(k)} = \|\tilde{q}_k\|\omega_j^{(k)}, \quad k \le j \le n,$

(4e) $\quad D^{(1)} = A,$

and X denotes a possibly nonzero element. Thus $A = \mathcal{U}^{(n)^{-T}} D^{(n)} \mathcal{U}^{(n)-1}$ is the Cholesky decomposition of A, since by (2) and (3) $\mathcal{U}^{(n)}$ and hence $\mathcal{U}^{(n)-1}$ are upper triangular with ones on the main diagonal. Because of this behavior in the quadratic case, we may regard Algorithm (1) as a nonlinear Cholesky decomposition method.

In order to bound $\|u_j^{(k)}\|$, we need to know something about $\|D^{(k)}\|$. Let $M^{(k)}$ denote the lower right-hand corner of $D^{(k)}$, so that

(5) $$D^{(k)} = \begin{bmatrix} \mathrm{diag}(d_1,\ldots,d_{k-1}) & 0 \\ 0 & M^{(k)} \end{bmatrix},$$

where $M^{(k)} = M^{(k)^T} \in \mathcal{L}(\mathbb{R}^{n-k+1})$.

(6) <u>Lemma</u>: If $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ is positive definite and (4) holds with $\mathcal{U}^{(k)}$ as above, i.e. upper triangular with ones on the main diagonal and I_{n-k+1} in the lower right-hand corner, then

(7a) $\quad \|D^{(k)}\| \le \|A\|$ and

(7b) $\quad \|D^{(k)-1}\| \le \|A^{-1}\|.$

<u>Proof</u>: By (4e) this holds for $k = 1$. Suppose $k \ge 2$ and that (7) holds for k replaced by $k-1$. Recall that the 2-norm of a positive

definite (symmetric) matrix equals its greatest eigenvalue, while the 2-norm of its inverse equals the reciprocal of its least eigenvalue. Note that any diagonal element of such a matrix lies between its least and greatest eigenvalues and any submatrix symmetric about the diagonal of the original one is also positive definite. Because of (4b,c) and (5), it thus suffices to show that $\|M^{(k)}\| \le \|A\|$ and $\|M^{(k)-1}\| \le \|A^{-1}\|$. Since $M^{(k)} = M^{(k)T}$, it suffices to show that

$$\|z\|^2/\|A^{-1}\| \le z^T M^{(k)} z \le \|A\|\|z\|^2 \qquad \text{for all} \quad z \in \mathbb{R}^{n-k+1}. \tag{8}$$

Now (4a) implies $A = \mathcal{U}^{(k)-T} D^{(k)} \mathcal{U}^{(k)-1}$ and $\mathcal{U}^{(k)-1}$ has the form

$$\mathcal{U}^{(k)-1} = \begin{bmatrix} V_1 & V_2 \\ 0 & I_{n-k+1} \end{bmatrix}, \text{ where } V_1 \in \mathcal{L}(\mathbb{R}^{k-1}) \text{ is (upper triangular and)}$$

nonsingular and $V_2 \in \mathcal{L}(\mathbb{R}^{n-k+1}, \mathbb{R}^{k-1})$. For any $z \in \mathbb{R}^{n-k+1}$ we thus have

$$\begin{aligned} \|A\|\|z\|^2 &\ge \begin{pmatrix} 0 \\ z \end{pmatrix}^T A \begin{pmatrix} 0 \\ z \end{pmatrix} = \begin{pmatrix} V_2 z \\ z \end{pmatrix}^T D^{(k)} \begin{pmatrix} V_2 z \\ z \end{pmatrix} \\ &= z^T M^{(k)} z + (V_2 z)^T \mathrm{diag}(d_1, \ldots, d_{k-1})(V_2 z) \\ &\ge z^T M^{(k)} z, \end{aligned}$$

which proves the second inequality in (8). Given $z \in \mathbb{R}^{n-k+1}$, let $y = -V_1^{-1} V_2 z$, so that $\mathcal{U}^{(k)-1}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ z \end{pmatrix}$, whence

$$z^T M^{(k)} z = \begin{pmatrix} y \\ z \end{pmatrix}^T A \begin{pmatrix} y \\ z \end{pmatrix} \ge \left\| \begin{pmatrix} y \\ z \end{pmatrix} \right\|^2 / \|A^{-1}\| \ge \|z\|^2/\|A^{-1}\|,$$

and the first inequality in (8) is also valid. ∎

In the quadratic case we may bound $\|u_j^{(k)}\|$ by rewriting (4a) as $A^{-1} = \mathcal{U}^{(k)T} D^{(k)^{-1}} \mathcal{U}^{(k)}$ and using (7a) to obtain

$$\|A^{-1}\| \geq e_j^T A^{-1} e_j = u_j^{(k)T} D^{(k)-1} u_j^{(k)} \geq \|u_j^{(k)}\|^2 / \|D^{(k)}\|$$

$$\geq \|u_j^{(k)}\|^2 / \|A\|,$$

whence $\|u_j^{(k)}\| \leq \sqrt{\|A\|\|A^{-1}\|} = \sqrt{\chi(A)}$.

Similar reasoning applies in the general case:

(9) <u>Lemma</u>: Suppose $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\phi'(x^*) = 0$ with $\phi''(x^*)$ positive definite. Assume $\lim_{h \to 0} [\sup \Xi(B(x^*,h))] = 0$. If $c_1 > \sqrt{\chi(\phi''(x^*))}$, then there exists $\rho > 0$ such that the $u_j^{(k)}$ generated by Algorithm (1) satisfy

(10) $$\|u_j^{(k)}\| \leq c_1 \quad \text{whenever} \quad y_0 = x_{i-1} \in B(x^*, \rho).$$

<u>Proof sketch</u>: WLOG no reordering occurs in (1b). Let $A = \phi''(x^*)$ and let $D^{(k)}$ be given by (4) when Algorithm (1) is applied to the problem $\phi(x) = \frac{1}{2}x^T A x$ with $\Xi \equiv 0$. For y_0 near x^* the $\mathcal{U}^{(k)}$ given by (3) when Algorithm (1) runs on the original data is such that $\tilde{D}^{(k)} \equiv \mathcal{U}^{(k)^T} A \mathcal{U}^{(k)}$ satisfies $\|D^{(k)} - \tilde{D}^{(k)}\| = \mathcal{O}(\xi)$, whence $\|D^{(k)-1} - \tilde{D}^{(k)-1}\| = \mathcal{O}(\xi)$. For small ξ we may use (7a) to obtain

$$\|A^{-1}\| \geq e_j^T A^{-1} e_j = e_j^T Q^{(k)T} \tilde{D}^{(k)-1} Q^{(k)} e_j$$

$$= u_j^{(k)T} D^{(k)-1} u_j^{(k)} - u_j^{(k)T} [D^{(k)-1} - \tilde{D}^{(k)-1}] u_j^{(k)}$$

$$\geq \|u_j^{(k)}\|^2 / \|D^{(k)}\| - \|u_j^{(k)}\|^2 \sigma(\xi)$$

$$\geq \|u_j^{(k)}\|^2 (1 - \sigma(\xi)) / \|A\|,$$

whence $\|u_j^{(k)}\| \leq \sqrt{\frac{\gamma(A)}{1-\sigma(\xi)}} = \sqrt{\frac{\gamma(\phi''(x^*))}{1-\sigma(\xi)}}$. Since

$\lim_{h\to 0} [\sup \Xi(\delta(x^*,h))] = 0$, there exists $\rho > 0$ such that (10) holds. ∎

As with Brown's method, if finite differences are used to approximate $\frac{\partial^2\phi}{\partial q_k \partial u_j^{(k)}}$ when computing $\omega_j^{(k)}$, it is probably best to approximate $\frac{\partial^2\phi}{\partial q_k \partial(u_j^{(k)}/\|u_j^{(k)}\|)}$ and note that $\frac{\partial^2\phi}{\partial q_k \partial u_j^{(k)}} = \|u_j^{(k)}\| \frac{\partial^2\phi}{\partial q_k \partial(u_j^{(k)}/\|u_j^{(k)}\|)}$.

This allows more accurate control of the differencing step size.

We now consider another way to choose p_k and $u_j^{(k)}$.

(11) Algorithm: [Orthogonal coordinate search];

(11a) Let the quantities described in (3.4.9a) and (3.3.8a) be given and run Algorithm (3.4.9) using (3.3.8) for ψ_1 and making the following choices;

(11b) In (3.4.9h) either proceed as in (1b) or let $p_1 = \bar{r}$;

(11c) To carry out (3.3.8c), let $u_1^{(1)},\ldots,u_n^{(1)}$ be an orthonormal basis for $\mathbb{R}^n$, e.g. $u_j^{(1)} = e_j$, and in place of (3.3.8e) do the following:

let $w_{k-1} = U_{k-1}\begin{pmatrix}\omega_{k-1}^{(k-1)}\\ \vdots \\ \omega_n^{(k-1)}\end{pmatrix} = \sum_{j=k-1}^{n} \omega_j^{(k-1)} u_j^{(k-1)}$, find an orthogonal

$\Omega \in \mathcal{L}(\mathbb{R}^{n-k+2})$ such that $\Omega\begin{pmatrix}\omega_{k-1}^{(k-1)}\\ \vdots \\ \omega_n^{(k-1)}\end{pmatrix} = \pm \|w_{k-1}\| e_1$ and let

$[u_{k-1}^{(k)}, u_k^{(k)}, \ldots, u_n^{(k)}] = U_{k-1}\Omega^T = [u_{k-1}^{(k-1)}, \ldots, u_n^{(k-1)}]\Omega^T$;

(11d) Choose $p_k = u_k^{(k)}$ in step (3.3.8f);

END Algorithm (11);

Note that the basis $u_k^{(k)}, \ldots, u_n^{(k)}$ is chosen as in Algorithm (2.4.1)--see step (2.4.1j). Thus the discussion in §2.4 about maintaining $u_k^{(k)}, \ldots, u_n^{(k)}$ is applicable here. In particular, step (11c) may be carried out by Powell's orthogonalization method, i.e. Algorithm (2.4.2), as well as by Householder or Givens transformations.

We readily see that the p_k and $u_j^{(k)}$ selected by Algorithm (11) meet the requirements of (3.3.8) and (3.4.9). Thus Theorem (3.4.24) applies.

Although the basis $u_k^{(k)}, \ldots, u_n^{(k)}$ maintained by Algorithm (11) will generally be somewhat better conditioned and should therefore give slightly better numerical performance than the one maintained by Algorithm (1), the overhead of maintaining this basis is higher, at least when the techniques considered in §2.4 are used.

Making a different choice of p_k, we obtain an algorithm which under the proper conditions produces minor and thus major iterates very close to those generated by several nonlinear conjugate gradient methods.

(12) Algorithm: [Projected gradient method];

(12a) Same as (11a);

(12b) Let $p_1 = \bar{f}$ in (3.4.9h);

(12c) Same as (11c);

(12d) To choose p_k in step (3.3.8f), compute an approximation to

$$P_{k-1}\nabla\phi(y_{k-1}) = U_k U_k^T \nabla\phi(y_{k-1}), \text{ e.g. } p_k = \sum_{j=k}^{n}\left(\frac{\Delta\phi}{\Delta u_j^{(k)}}(y_{k-1},h)\right)u_j^{(k)};$$

END Algorithm (12);

Again we readily see that the requirements of (3.3.8) and (3.4.9) regarding p_k and $u_j^{(k)}$ are met, whence Theorem (3.2.24) applies.

Algorithm (12) requires the same overhead as Algorithm (11) to maintain $u_k^{(k)},\ldots,u_n^{(k)}$. If p_k is computed as $\sum_{j=k}^{n}\left(\frac{\Delta\phi}{\Delta u_j^{(k)}}(y_{k-1},h)\right)u_j^{(k)}$ and $\omega_j^{(k)}$ is computed as $\frac{\Delta^2\phi}{\Delta q_k \Delta u_j^{(k)}}(y_{k-1},h)$ or $\frac{\Delta^2\phi}{\Delta p_k \Delta u_j^{(k)}}(y_{k-1},h)$, then the function values used in determining p_k may also be used in determining $\omega_j^{(k)}$, $k \le j \le n$, so choosing p_k in Algorithm (12) need not be as expensive as might first appear. If p_k is computed as just suggested, then it will (usually) be necessary to spend at least two more function evaluations to compute a sufficiently accurate value of $\bar{f}_k$, say $\bar{f}_k = \frac{\tilde{\Delta}\phi}{\Delta q_k}(y_{k-1},h)$, i.e. by central differences. Thus Algorithm (12) may cost about 2n more function evaluations per complete major iteration than Algorithm (11), though, of course, both require $O(n^2)$. Moreover, if ψ_1 returns early in step

(3.3.8σ), then the extra function evaluations needed to determine the last (unsuccessful) p_k will have been performed in vain.

Consider how Algorithm (12) behaves when $\phi(x) = \frac{1}{2}x^TAx - b^Tx$ is quadratic with $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ positive definite and all calculations are exact. We shall see below (in the proof of Theorem (14)) that $q_k = \pm \nabla\phi(y_{k-1})/\|\nabla\phi(y_{k-1})\| = \pm r_k/\|r_k\|$, where $r_j = \nabla\phi(y_{j-1}) = Ay_{j-1} - b$ denotes the $j^{\underline{th}}$ residual. In this respect Algorithm (12) strongly resembles Craig's method (§2.5). Thus it should come as no great surprise that in the present exact quadratic case Algorithm (12) produces the same sequence of minor iterates (in the first and only nontrivial major iteration) as the conjugate gradient method (Faddeev and Faddeeva, 1963, §69.3), to which Craig's method is closely related (see Faddeev and Faddeeva). The conjugate gradient method solves the linear system $\nabla\phi(x) = Ax - b = 0$ as follows.

(13) Algorithm: [Linear conjugate gradient method for solving $Ax^* = b$];

(13a) Let $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ with A positive definite and $b, \hat{y}_0 \in \mathbb{R}^n$ be given;

(13b) $\hat{p}_1 \leftarrow \hat{r}_1 \leftarrow A\hat{y}_0 - b$;

LOOP:

DO k = 1 TO n WHILE $(\hat{r}_k \neq 0)$;

(13c) $$\hat{y}_k \leftarrow \hat{y}_{k-1} - \left(\frac{\hat{p}_k^T \hat{r}_k}{\hat{p}_k^T A \hat{p}_k}\right)\hat{p}_k;$$

(13d) $$\hat{r}_{k+1} \leftarrow A\hat{y}_k - b = \hat{r}_k - \left(\frac{\hat{p}_k^T \hat{r}_k}{\hat{p}_k^T A \hat{p}_k}\right) A\hat{p}_k;$$

(13e) $\hat{p}_{k+1} \leftarrow \hat{r}_{k+1} - \left(\frac{\hat{r}_{k+1}^T A\hat{p}_k}{\hat{p}_k^T A\hat{p}_k}\right)\hat{p}_k = \hat{r}_{k+1} + \left(\frac{\|\hat{r}_{k+1}\|^2}{\|\hat{r}_k\|^2}\right)\hat{p}_k;$

END LOOP;

Return $(x^* = \hat{y}_{k-1})$;

END Algorithm (13);

(14) Theorem: Let $A = A^T \in \mathcal{L}(\mathbb{R}^n)$ be positive definite and let $b, x_0 \in \mathbb{R}^n$ be given. Run Algorithm (13) with $\hat{y}_0 = x_0$, and if $r_k = 0$ for some $k \leq n$, then let $\hat{y}_j = \hat{y}_{k-1}$ for $k \leq j \leq n$. If Algorithm (12) runs with $\phi(x) = \frac{1}{2} x^T Ax - b^T x$ and does exact calculations, then the minor iterates $y_0, \ldots, y_n$ produced during the first major iteration satisfy $y_j = \hat{y}_j$, $0 \leq j \leq n$.

Proof: Let $r_j = \nabla\phi(y_{j-1}) = Ay_{j-1} - b$. The search directions p_k chosen by Algorithm (12) are then $p_k = P_{k-1}r_k$. In particular, $y_0 = \hat{y}_0 = x_0$, $p_1 = \hat{p}_1 = \hat{r}_1 = r_1$, and $q_1 = p_1/\|p_1\|$ is proportional to r_1.

Suppose inductively that $1 \leq k \leq n$, $y_{k-1} = \hat{y}_{k-1}$, q_j is proportional to r_j, and p_j is proportional to $\hat{p}_j$ with

(15) $$p_j, \hat{p}_j, \hat{r}_j = r_j \in \mathrm{Span}\{r_1, Ap_1, \ldots, Ap_{j-1}\}, \quad 1 \leq j \leq k.$$

Since $\bar{r}_k = \frac{\partial\phi}{\partial q_k}(y_{k-1}) = \|\tilde{q}_k\|^{-1} \frac{\partial\phi}{\partial p_k}(y_{k-1}) = p_k^T r_k/\|\tilde{q}_k\|$ and

$\pi_k = \frac{\partial^2\phi}{\partial q_k \partial p_k} = \|\tilde{q}_k\|^{-1} \frac{\partial^2\phi}{\partial p_k^2} = p_k^T Ap_k/\|\tilde{q}_k\|$, we have

$\frac{\bar{f}_k}{\pi_k} p_k = \left(\frac{p_k^T r_k}{p_k^T A p_k}\right) p_k = \left(\frac{\hat{p}_k^T \hat{r}_k}{\hat{p}_k^T A \hat{p}_k}\right) \hat{p}_k$, whence $y_k = \hat{y}_k$. From (13 d,e) and (15) we see that $\hat{p}_{k+1}, \hat{r}_{k+1} \in \text{Span}\{r_1, Ap_1, \ldots, Ap_k\}$. WLOG $\hat{r}_{k+1} \neq 0$. Since, by inductive assumption, $\hat{r}_j = r_j$ is proportional to q_j, and since

$r_{k+1}^T r_j = \pm\|r_j\|\phi'(y_k)q_j = \pm\|r_j\|f_j(y_k) = 0, \quad 1 \leq j \leq k, \quad r_i^T r_j = 0 \text{ for } i \neq j$

(see (Faddeev and Faddeeva, 1963, §69.3) for an alternative proof of this fact). Thus $r_1, \ldots, r_{k+1}$ are linearly independent, whence by (15) we have $\text{Span}\{r_1, \ldots, r_k\} = \text{Span}\{r_1, Ap_1, \ldots, Ap_{k-1}\}$. Together with (13d) and the inductive assumption that p_k is proportional to $\hat{p}_k$, this implies that Ap_k is linearly independent of $r_1, Ap_1, \ldots, Ap_{k-1}$ (since otherwise $r_{k+1}^T r_j = 0$ for $1 \leq j \leq k$ would imply $r_{k+1} = \hat{r}_{k+1} = 0$). By (13d) $r_{k+1} = \hat{r}_{k+1}$ is a linear combination of r_1 and Ap_j, $1 \leq j \leq k$, with the r_1 coefficient $= 1 \neq 0$, so $\text{Span}\{r_1, Ap_1, \ldots, Ap_k\} = \text{Span}\{r_{k+1}, Ap_1, \ldots, Ap_k\}$ and p_k is a linear combination of r_{k+1} and $Ap_1, \ldots, Ap_k$ by (15). Since $p_{k+1}^T Ap_j = \hat{p}_{k+1}^T A\hat{p}_j = 0$, $1 \leq j \leq k$, we see from (13e) that $\hat{p}_{k+1}$ is proportional to the orthogonal projection $P_k r_{k+1}$ of r_{k+1} onto

$$\begin{aligned} \text{Span}\{Ap_1, \ldots, Ap_k\}^{\perp} &= \text{Span}\{Aq_1, \ldots, Aq_k\}^{\perp} \\ &= \text{Span}\{w_1, \ldots, w_k\}^{\perp}. \end{aligned}$$

But $p_{k+1} = P_k r_{k+1}$, so p_{k+1} is proportional to $\hat{p}_{k+1}$.

Note from (3.3.8g,h) that p_k and hence $\hat{p}_k$ lie in $\text{Span}\{q_1, \ldots, q_k\}$. As seen above, $q_j^T r_{k+1} = 0$, $1 \leq j \leq k$, so $Q_k^+ \hat{r}_{k+1} = Q_k^T r_{k+1} = 0$, where $Q_k = [q_1, \ldots, q_k]$. From (13e) and the proportionality of p_{k+1} to $\hat{p}_{k+1}$

we thus see that q_{k+1} is proportional to $(I-Q_kQ_k^+)\hat{p}_{k+1} = \hat{r}_{k+1} = r_{k+1}$. This completes the inductive step. ∎

Several people have devised generalizations of Algorithm (13) for minimizing nonquadratic functions ϕ. Cohen (1972) described the methods of Fletcher and Reeves (1964), Daniel (1967), and Polak and Ribière (1969), all with reinitialization, and proved that they are n_1-step Q-quadratically convergent under reasonable conditions, such as $\phi \in C^3(\mathbb{R}^n, \mathbb{R})$, where $n_1 \geq n$ is the number of steps between reinitializations. Following Cohen, we designate these methods FR, D, and PR respectively. They have the form:

(16) <u>Algorithm</u>: [generalized conjugate gradient methods];

(16a) Let $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$, $\hat{x}_0 \in \mathbb{R}^n$, and $n_1 \geq n$ be given;

OUTERLOOP:

DO $i = 1$ TO ∞;

(16b) $\hat{y}_0 \leftarrow \hat{x}_{i-1}$;

(16c) $\hat{p}_1 \leftarrow \hat{r}_1 \leftarrow \nabla\phi(\hat{y}_0)$;

INNERLOOP:

DO $k = 1$ TO n_1;

(16d) If $\hat{r}_k = 0$ then stop and let $\hat{y}_k = \ldots = \hat{y}_n = \hat{x}_j = \hat{y}_{k-1}$ for all $j \geq i$;

(16e) $\hat{y}_k \leftarrow \hat{y}_{k-1} - \hat{\lambda}_k\hat{p}_k$, where $\hat{\lambda}_k \geq 0$ is such that

$$\phi(\hat{y}_{k-1}-\hat{\lambda}_k\hat{p}_k) = \min\{\phi(\hat{y}_{k-1}-\lambda\hat{p}_k) \mid \lambda \in [0,\infty)\};$$

(16f) $\hat{r}_{k+1} \leftarrow \nabla\phi(\hat{y}_k)$;

Either

$$(16g)_{FR} \qquad \hat{p}_{k+1} \leftarrow \hat{r}_{k+1} + \frac{\|\hat{r}_{k+1}\|^2}{\|\hat{r}_k\|^2}\hat{p}_k$$

or

$$(16g)_{D} \qquad \hat{p}_{k+1} \leftarrow \hat{r}_{k+1} - \left(\frac{\hat{r}_{k+1}^T \emptyset''(\hat{y}_k)\hat{p}_k}{\hat{p}_k^T \emptyset''(\hat{y}_k)\hat{p}_k}\right)\hat{p}_k$$

or

$$(16g)_{PR} \qquad \hat{p}_{k+1} \leftarrow \hat{r}_{k+1} + \left(\frac{\hat{r}_{k+1}^T(\hat{r}_{k+1}-\hat{r}_k)}{\|\hat{r}_k\|^2}\right)p_k;$$

END INNERLOOP;

$$(16h) \qquad \hat{x}_i \leftarrow \hat{y}_{n_1}$$

END OUTERLOOP and Algorithm (16);

All three of the choices for (16g) lead to similar behavior near a nonsingular minimizer x^* (at least when $\emptyset \in C^3(\mathbb{R}^n, \mathbb{R})$): Cohen shows that the minor iterates $\hat{y}_j^{(FR)}$, $\hat{y}_j^{(D)}$, and $\hat{y}_j^{(PR)}$ satisfy $\|\hat{y}_j^{(FR)} - \hat{y}_j^{(D)}\|, \|\hat{y}_j^{(FR)} - \hat{y}_j^{(PR)}\| = \mathcal{O}(\|\emptyset'(\hat{y}_0)\|^2)$, assuming $\hat{y}_0^{(FR)} = \hat{y}_0^{(D)} = \hat{y}_0^{(PR)} = \hat{y}_0$. We can probably make a similar statement involving the minor iterates produced by Algorithm (12), at least when $\|p_k - P_{k-1}\nabla\emptyset(y_{k-1})\| = \mathcal{O}(\|\emptyset'(y_0)\|^2)$:

(17) <u>Conjecture</u>: Assume $\emptyset \in C^3(\mathbb{R}^n, \mathbb{R})$ and $x^* \in \mathbb{R}^n$ satisfy $\emptyset'(x^*) = 0$ with $\emptyset''(x^*)$ positive definite. Let Algorithm (12) be executed

with $\|p_k - P_{k-1}\nabla\emptyset(y_{k-1})\| = \mathcal{O}(\|\emptyset'(y_{k-1})\|^2)$. If the major iterates $\hat{x}_0, \hat{x}_1, \ldots$ produced by Algorithm (16) converge to x^* and if Algorithm (12) runs with $\Xi(x) = \mathcal{O}(\|\emptyset'(y_0)\|^2)$ near x^*, then the minor iterates y_k and $\hat{y}_k$ produced by (12) and (16) satisfy $\|y_k - \hat{y}_k\| = \mathcal{O}(\|\emptyset'(y_0)\|^2)$ whenever $y_0 = \hat{y}_0$ and $\|y_0 - x^*\|$ is sufficiently small.

Probable proof sketch: Let the p_k used by (12) be scaled so that $\|p_k\| = 1$. (This spares us a factor of $\|p_k\|$ in certain estimates but otherwise changes nothing.) Let $\lambda_k = \bar{f}_k/\pi_k$ denote the step length taken in (12), and let $h = \|\nabla\emptyset(y_0)\|$. For $y_0 = \hat{y}_0$ close to x^*, we find $\|y_1 - \hat{y}_1\| = \|\lambda_1 p_1 - \hat{\lambda}_1\hat{p}_1\| = \mathcal{O}(h^2)$. Inductively assume $\|y_k - \hat{y}_k\| = \mathcal{O}(h^2)$ and $\|\lambda_k p_k - \hat{\lambda}_k\hat{p}_k\| = \mathcal{O}(h^2)$. As Cohen (1972) shows, $|\hat{\lambda}_k| = \mathcal{O}(1)$, so $\left\|\frac{\lambda_k}{\hat{\lambda}_k}p_k - \hat{p}_k\right\| = \mathcal{O}(h^2)$. Let

$$\bar{p}_{k+1} = \nabla\emptyset(y_k) - \left(\frac{\emptyset'(y_k)\emptyset''(y_k)p_k}{p_k^T\emptyset''(y_k)p_k}\right)p_k;$$

then $\bar{p}_{k+1}^T\emptyset''(y_k)p_k = 0$ and $|\bar{p}_{k+1}\emptyset''(y_k)p_j| = \mathcal{O}(h^2)$, $1 \le j \le k$, since $|\emptyset'(y_k)\emptyset''(y_k)p_j| = \mathcal{O}(h^2)$, $\|\emptyset'(y_k)\| = \mathcal{O}(h)$, and $|p_k^T\emptyset''(y_k)p_j| = \mathcal{O}(h)$. Now we should be able to show that the projection of p_{k+1} onto $\mathrm{Span}\{\nabla\emptyset(y_k), \emptyset''(y_k)p_1, \ldots, \emptyset''(y_k)p_k\}^{\perp}$ has norm $\le \mathcal{O}(h)$ and then that

$$\|\bar{p}_{k+1} - \|\bar{p}_{k+1}\|p_{k+1}\| = \mathcal{O}(h^2). \tag{18a}$$

From Cohen's results (1972) we may assume WLOG that $(16g)_D$ is used, so

(18b) $$\|\hat{p}_{k+1} - \bar{p}_{k+1}\| \le \|\nabla\phi(\hat{y}_k) - \nabla\phi(y_k)\| +$$

$$+ \left| \frac{\phi'(y_k)\phi''(y_k)p_k}{p_k^T\phi''(y_k)p_k} - \left(\frac{\lambda_k}{\hat{\lambda}_k}\right)\left(\frac{\phi'(\hat{y}_k)\phi''(\hat{y}_k)\hat{p}_k}{\hat{p}_k^T\phi''(\hat{y}_k)\hat{p}_k}\right)\right| \|p_k\| +$$

$$+ \left\| \frac{\lambda_k}{\hat{\lambda}_k} p_k - \hat{p}_k \right\| \left| \frac{\phi'(\hat{y}_k)\phi''(\hat{y}_k)\hat{p}_k}{\hat{p}_k^T\phi''(\hat{y}_k)\hat{p}_k} \right|$$

$$= \mathcal{O}(h^2).$$

Let $\bar{\lambda}_{k+1} = \dfrac{\phi'(y_k)\bar{p}_{k+1}}{\bar{p}_{k+1}^T\phi''(y_k)\bar{p}_{k+1}}$ and $\bar{y}_{k+1} = y_k - \bar{\lambda}_{k+1}\bar{p}_{k+1}$: from (18) we find $\|y_{k-1} - \bar{y}_{k+1}\| = \mathcal{O}(h^2)$ and $\|\bar{y}_{k+1} - \hat{y}_{k+1}\| = \mathcal{O}(h^2)$, whence $\|y_{k+1} - \hat{y}_{k+1}\| = \mathcal{O}(h^2)$; furthermore, $\|\lambda_{k+1}p_{k+1} - \bar{\lambda}_{k+1}\bar{p}_{k+1}\| = \mathcal{O}(h^2)$ and $\|\bar{\lambda}_{k+1}\bar{p}_{k+1} - \hat{\lambda}_{k+1}\hat{p}_{k+1}\| = \mathcal{O}(h^2)$, whence $\|\lambda_{k+1}p_{k+1} - \hat{\lambda}_{k+1}\hat{p}_{k+1}\| = \mathcal{O}(h^2)$. This completes the inductive step.

∎

(19) <u>Conjecture</u>: If the hypothesis $\phi \in C^3(\mathbb{R}^n, \mathbb{R})$ of (17) is changed to: $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and there exist $\mu \in (0,1]$, $\kappa \in [0,\infty)$, and $\rho_2 \in (0,\infty]$ such that $\|\phi''(x) - \phi''(y)\| \le \kappa\|x-y\|^\mu$ for all $x,y \in B(x^*,\rho_2)$, then the conclusion of Conjecture (17) holds with $\|y_k-\hat{y}_k\| = \mathcal{O}(\|\phi'(y_0)\|^2)$ changed to $\|y_k-\hat{y}_k\| = \mathcal{O}(\|\phi'(y_0)\|^{1+\mu})$. ∎

Note that Conjecture (17) does not hold if p_k satisfies only $\|p_k-P_{k-1}\nabla\phi(y_{k-1})\| = \mathcal{O}(\|\phi'(y_0)\|)$, as would probably be the case if p_k were computed as

$$p_k = \sum_{j=k}^{n} \left(\frac{\Delta\phi}{\Delta u_j^{(k)}}(y_{k-1},h)\right) u_j^{(k)}.$$

However, there appears to be no special reason why p_k should be computed more accurately than this in Algorithm (12).

Rigorously proved, Conjecture (17) would provide an alternate proof of the Q-quadratic convergence of the several generalized conjugate gradient methods (16), and Conjecture (19) would allow us to make a slightly more general statement about these methods.

The analogy between Algorithms (16) and (12) suggests that we could replace the exact gradient required in step (16f) by an approximation $\hat{r}_k$ satisfying $\|\hat{r}_k - \nabla\phi(\hat{y}_{k-1})\| = \mathcal{O}(\|\phi'(\hat{y}_0)\|^2)$, e.g. $\hat{r}_k = \left(\frac{\tilde{\Delta}\phi}{\Delta e_1}(y_{k-1},h),\ldots,\frac{\tilde{\Delta}\phi}{\Delta e_n}(y_{k-1},h)\right)^T$ with $h = \mathcal{O}(\|\phi'(\hat{y}_0)\|)$. The analogy also suggests that the exact line search requested in (16e) could be replaced by a single finite difference Newton step if this yields a sufficient decrease in ϕ, or else a step of length chosen by, say, the Armijo technique. Let (16)' denote the finite difference form of (16) which we thus obtain: Cohen's (1972) proof should be extendable to show local Q-quadratic convergence of (16)'. Moreover, I strongly suspect that Conjectures (17) and (19) remain valid for (16) replaced by (16)'; note for (19) that the finite difference approximation $\hat{r}_k$ to $\nabla\phi(\hat{y}_{k-1})$ would only need to satisfy $\|\hat{r}_k - \nabla\phi(\hat{y}_{k-1})\| = \mathcal{O}(\|\phi'(\hat{y}_0)\|^{1+\mu})$. Near x^* a single finite difference Newton step should always suffice in step (16e)'.

Computational evidence--in the form of an APL program which computes $\hat{r}_k$ by forward differences with differencing step size

$h = \mathcal{O}(\|\emptyset'(y_0)\|$ but otherwise runs Algorithm (16) with quadratic $\emptyset$--strongly suggests that (16)' must compute $\hat{r}_k$ with $\|\hat{r}_k - \nabla\emptyset(\hat{y}_{k-1})\| = \mathcal{O}(\|\emptyset'(\hat{y}_0)\|^2)$ and not just $\|\hat{r}_k - \nabla\emptyset(\hat{y}_{k-1})\| = \mathcal{O}(\|\emptyset'(\hat{y}_0)\|)$. Thus central differences are called for, unless arithmetic of sufficient accuracy is available to compute accurate forward differences with step size $h = \mathcal{O}(\|\emptyset'(y_0)\|^2)$. If central differences are used, then (16)' requires nearly twice as many function evaluations per complete major iteration as Algorithm (12), (11), or (1) but enjoys no greater rate of convergence. (If $(16g)'_D$ is used, then (16)' requires extra function evaluations to compute $\hat{p}_{k+1}$.) Similarly, if (16) is executed and the cost of computing $\nabla\emptyset$ exceeds the cost of performing roughly n evaluations of $\emptyset$, then (12), (11), or (1) requires less work per major iteration than (16) but enjoys the same convergence rate. Of course, (16) and (16)' require less overhead than (1), (11), or (12) and may perform differently far from x^*.

We may consider implementing algorithms (1), (11), and (12) in finite but variable precision arithmetic, just as we did Brown's method in §2.3. Assume $\emptyset \in C^2(\mathbb{R}^n, \mathbb{R})$ has a nonsingular minimizer x^* near which $\emptyset''$ satisfies a Lipschitz condition of order $\mu \in (0,1]$. Suppose we compute $\bar{r}_k = fl\left(\frac{\tilde{\Delta}\emptyset}{\Delta q_k}(y_{k-1},h)\right)$, $\pi_k = fl\left(\frac{\Delta^2\emptyset}{\Delta q_k \Delta p_k}(y_{k-1},h)\right)$, and $\omega_j^{(k)} = fl\left(\frac{\Delta^2\emptyset}{\Delta q_k \Delta u_j^{(k)}}(y_{k-1},h)\right)$, with $h = \mathcal{O}(\|\emptyset'(y_0)\|)$. Reasoning similar to that used in Lemma (2.3.12) shows that we may effect these calculations with accuracy factor $\xi = \mathcal{O}(\|\emptyset'(y_0)\|^{\mu})$ near x^* by performing them in floating-point arithmetic with unit roundoff $\lambda_3 = \mathcal{O}(\|\emptyset'(y_0)\|^{2+\mu})$. The same arithmetic may be used to compute $\bar{y}_k$ (3.3.8m), though in the exceptional case when $\mu < 1$ arithmetic with unit roundoff of order

$\sigma(\|\phi'(y_0)\|^{1+2\mu})$ suffices. To perform the housekeeping arithmetic of maintaining $u_k^{(k)},\ldots,u_n^{(k)}$, we may achieve an accuracy factor of $\xi = \sigma(\|\phi'(y_0)\|^{\mu})$ by using arithmetic with unit roundoff $\lambda_1 = \sigma(\|\phi'(y_0)\|^{\mu})$. Thus, to use finite-precision arithmetic without suffering a reduction in the rate of convergence, the minimization algorithms proposed here require only "single" precision arithmetic for the housekeeping operations, but require "triple" precision for the remaining operations. By way of contrast, Brown's method required only "double" precision for the nonhousekeeping operations. The minimization algorithms require an extra degree of arithmetic precision for the finite differences because they approximate $\underline{f} = \nabla\phi$ itself by finite differences.

A slightly different line of reasoning would show that step (3.3.8m) could be changed to read

$$\text{"Pick } \bar{y}_k \in \bar{B}(y_{k-1} - \frac{\breve{f}_k}{\pi_k} p_k,\ \epsilon_9 \xi \|\phi'(y_0)\|);\text{" ,}$$

i.e. with the exponent $1+\mu$ removed from $\|\phi'(y_0)\|$. Only such accuracy is necessary, since near x^* we have $\phi(\bar{y}_k) > \phi(y_{k-1})$ only if $|f_k(\bar{y}_k)| = |q_k^T \nabla\phi(\bar{y}_k)| > |f_k(y_{k-1})|$, in which case we incur less error by choosing $y_k = y_{k-1}$ than $y_k = \bar{y}_k$, despite the fact that (3.2.6ℓ) might be violated. Thus "double" precision suffices for determining $\bar{y}_k$, just as it sufficed for determining y_k in Brown's method.

3.6 Numerical Tests and Possible Improvements

We now consider how to implement Algorithms (3.5.1, 11, 12) in practice on a machine with arithmetic of finite, fixed precision. We describe some preliminary numerical experience with such an implementation and suggest modifications that might lead to improved performance.

Assume that $\bar{f}_k$, π_k, $\omega_j^{(k)}$ are computed by

$$\bar{f}_k = fl\left(\frac{\tilde{\Delta}\phi}{\Delta q_k}(y_{k-1},h)\right),\ \pi_k = fl\left(\frac{\Delta^2\phi}{\Delta q_k \Delta p_k}(y_{k-1},h)\right),\ \omega_j^{(k)} = fl\left(\frac{\Delta^2\phi}{\Delta q_k \Delta u_j^{(k)}}(y_{k-1},h)\right),$$

possibly with q_k replaced by $p_k/\|\tilde{q}_k\|$ in some or all of these expressions. To achieve accurate control of the differencing step size, it is probably wise to scale p_k and $u_j^{(k)}$ to keep $\|p_k\|$ and $\|u_j^{(k)}\|$ near 1. (Note in Algorithms (3.5.11,12) that $\|u_j^{(k)}\| = 1$ already, while the same updating formula may be used in Algorithm (3.5.1) if $u_j^{(k)}$ is replaced by $u_j^{(k)}/\|u_j^{(k)}\|$, $k \leq j \leq n$, before $\omega_j^{(k)}$ and $u_j^{(k+1)}$ are computed.)

Given that we must use floating-point arithmetic of fixed, finite precision, it is probably best to choose h in an effort to compute $\bar{f}_k$, π_k, $\omega_j^{(k)}$ as accurately as possible. Perhaps the most crucial calculation is that of $\bar{f}_k$, so we would probably do well choosing h to optimize the accuracy of $\bar{f}_k$. Assume ϕ satisfies a Lipschitz condition of order 1 and that we have an estimate $\bar{\kappa}$ for the Lipschitz constant κ. (Assuming $\phi \in C^3(\mathbb{R}^n, \mathbb{R})$, we might choose $\bar{\kappa}$ to be an upper bound on $\|\phi'''(x)\|$ for x in the region of interest.) Suppose also that we have an estimated upper bound $E = E(y_{k-1})$ on the absolute error $|\phi(y_{k-1}) - fl(\phi(y_{k-1}))|$ in the computed value of $\phi(y_{k-1})$. The error $\left|\frac{\tilde{\Delta}\phi}{\Delta q_k}(y_{k-1},h) - fl\left(\frac{\tilde{\Delta}\phi}{q_k}(y_{k-1},h)\right)\right|$ should then be bounded by $\frac{2E}{2h} = \frac{E}{h}$. On the other hand, (1.2.16) implies that $\frac{\bar{\kappa}h^2}{2}$ should bound $\left|\frac{\partial\phi}{\partial q_k}(y_{k-1}) - \frac{\tilde{\Delta}\phi}{\Delta q_k}(y_{k-1},h)\right|$. Thus $\frac{E}{h} + \frac{\bar{\kappa}h^2}{2}$ should bound the total (absolute) error in $\bar{f}_k$. This bound is minimized at $h = (E/\bar{\kappa})^{1/3}$. The preliminary test program computes such an h once per major iteration under the assumption that $E = E(y_0) = E_0 + E_1|\phi(y_0)|$ bounds the absolute error in $fl(\phi(y_0))$, where the constants E_0 and E_1 are given as parameters. This strategy seems to give quite satisfactory performance.

Algorithms (3.5.1,11,12) require that an approximation $\bar{f}$ to $\phi(y_0)$ be computed once per major iteration. So long as $\|\phi'(y_0)\|$ is large, it suffices to compute $\bar{f}$ by forward differences. But when $\|\phi'(y_0)\|$ is small, we should probably use central differences to compute $\bar{f}$ with good relative accuracy. The preliminary test program computes an initial $\bar{f}$ by forward differences; if $\|\bar{f}\|$ is less than a certain computed bound, then $\bar{f}$ is recomputed using central differences. (The actual test is whether $\|\bar{f}\|^2$ is less than a certain parameter times $\max\{(h\sigma)^2, (2E/h)^2\}$, where E is as above and σ is an estimate of the maximum eigenvalue of $\phi''(y_0)$; the estimate σ is computed during the previous major iteration as $\sigma = \max\{\sigma_0,\pi_1\}$ for Algorithm (3.5.1) and $\sigma = \max\{\sigma_0,\pi_1,\|w_1\|,\ldots,\|w_{n-1}\|\}$ for (3.5.11,12), with p_1 scaled so that $\|p_1\| = 1$ and where σ_0 is a parameter.) During preliminary testing $\bar{f}$ was usually computed only by forward differences until the last couple of major iterations.

Various stopping criteria are possible, e.g. whether $\|\bar{f}\|$, $\|x_i - x_{i-1}\|$, or $\phi(x_{i-1}) - \phi(x_i)$ are sufficiently small. Perhaps the most noteworthy stopping criterion in the preliminary test program is derived as follows. With h chosen as above, $\bar{\kappa}h^2$ should bound the error in $\bar{f}_k$.

Thus if $\|\bar{f}\|/(\bar{\kappa}h^2)$ is small, then the arithmetic accuracy available is probably no longer sufficient to allow a significant decrease in $\|x_i - x^*\|$, so further effort would probably be wasted. The criterion this suggests, i.e. whether $\|\bar{f}\|^2$ is less than a parameter times $(\bar{\kappa}h^2)^2$, frequently stopped the preliminary tests.

The preliminary test program, written in PL/I with all floating variables declared FLOAT(16) and run on Cornell University's IBM 360/65, implements Algorithms (3.5.1,11,12) with h, $\bar{f}$, $\bar{f}_k$, π_k, $\omega_j^{(k)}$ computed as described above and with various options possible for some of the calculations. One of these options is to replace q_k by $p_k/\|q_k\|$ in computing $\bar{f}_k$ and π_k or $\bar{f}_k$, π_k, and $\omega_j^{(k)}$. Such replacement seemed to give slightly better performance in the few runs made both with and without it. Another option is to use y_k in place of y_{k-1} in computing $\omega_j^{(k)}$, i.e. $\omega_j^{(k)} = fl\left(\frac{\Delta^2\phi}{\Delta q_k \Delta u_j^{(k)}}(y_k, h)\right)$, as suggested in Lemma (3.2.31). This gave sometimes better, sometimes worse performance. Yet another option is to reuse p_k and π_k one or more times as discussed in §2.6. In a few comparisons with ϕ = EXP3 (defined below), one reuse gave sometimes better, sometimes worse performance.

The choices available for ϕ in the preliminary tests included the functions listed by Dixon (1973) (as well as a quadratic function for debugging purposes). The choices which received the most attention were Rosenbrock's function $\phi \in C^\infty(\mathbb{R}^2, \mathbb{R})$, given by

$$\phi(x^{(1)}, x^{(2)}) = 100([x^{(1)}]^2 - x^{(2)})^2 + (1-x^{(1)})^2$$

with $x_0 = (-1.2, 1)^T$, and the function ϕ = EXP3 $\in C^\infty(\mathbb{R}^3, \mathbb{R})$, given by

$$\phi(x^{(1)},x^{(2)},x^{(3)}) = \sum_{j=1}^{10} [\exp(-jx^{(1)}/10) - x^{(3)}\exp(-jx^{(2)}/10)$$

$$- \exp(-j/10) + 5\exp(-j)]^2$$

with $x_0 = (1,2,1)^T$. The former was typically reduced to something less than 10^{-20}, while the latter was typically reduced to around 10^{-15}. In general, the region of fast convergence seemed to have a very small diameter for the nonquadratic test functions. Performance was rather disappointing in that over twice as many function evaluations were required by Algorithms (3.5.1,11,12) than the best results reported by Dixon (1973). (However, Dixon's stopping criteria are not clear.) The least number of function evaluations achieved for Rosenbrock's function was 227, while the least for EXP3 was 141.

No one of Algorithms (3.5.1,11,12) did consistently better or consistently worse than the other two in the preliminary tests. At this point it is thus unclear which of Algorithms (3.5.1,11,12) holds the most promise or which of the above described options it is generally best to exercise. It is also unclear what choices of the sundry control constants (e.g. $c_2,\ldots,c_6$) might be best, since very few changes in these were made in the preliminary tests.

What is clear is that faster global convergence would be highly desirable. Other choices of p_1, perhaps by a quasi-Newton method, might be worthwhile. Analogy with Rosenbrock's method (Rosenbrock, 1960) (see also (Powell, 1968)) suggests that choosing $p_1 = x_{i-1} - x_{i-2}$ or $p_1 = x_{i-1} - x_{i-3}$ (provided $|p_1^T \bar{f}| \geq \theta_2 \|p_1\| \|\bar{f}\|$ for some fixed $\theta_2 > 0$) might be of value. As suggested in §2.6, performance might also be improved if damped steps are taken in steps (3.3.8m) and (3.4.9m); the damping factor d_i could be increased toward 1 after a major iteration in which ψ_1, i.e. Algorithm (3.3.8), runs to completion, and decreased if ψ_1 returns in step (3.3.8o) because of an insufficient reduction in $\emptyset$. Or perhaps the third test in (3.3.8n) should be made less stringent.

The finite difference Newton's method requires but $n^2/2 + \mathcal{O}(n)$ function evaluations per (major) iteration, while the implementations described here of the proposed minimization algorithms use $n^2 + \mathcal{O}(n)$. We may cut this latter number roughly in half for Algorithms (3.5.1,11) by computing all the $\omega_j^{(k)}$ at one point, either y_0 or y_1, choosing p_k properly, and using p_k in place of q_k when computing the $\omega_j^{(k)}$. In Algorithm (3.5.1) p_k has at most k nonzero components. If Powell's orthogonalization method (2.4.2) is used to maintain $u_k^{(k)},\ldots,u_n^{(k)}$ in Algorithm (3.5.11), then the choice $p_k = u_n^{(k)}$ (rather than $p_k = u_k^{(k)}$) again results in p_k having at most k nonzero components. With such p_k we need only have approximations to (the first) k rows of $\emptyset''(y)$ to compute

$\omega_j^{(k)} \approx \dfrac{\partial^2\emptyset}{\partial p_k \partial u_j^{(k)}}(y) = p_k^T\emptyset''(y)u_j^{(k)}$, with $y = y_0$ or y_1. We may take advantage of the symmetry of $\emptyset''$ to thus compute the $\omega_j^{(k)}$ at a cost of only $n^2/2 + \mathcal{O}(n)$ function evaluations per major iteration when the elements of $\emptyset''(y)$ are approximated as $\dfrac{\Delta^2\emptyset}{\Delta e_i \Delta e_j}(y,h)$. Using Lemma (2.1.20), we see that (3.3.8d) is still satisfied when the $\omega_j^{(k)}$ are computed in this way.

Computational experiments must decide which, if any, of the modifications suggested above are worthwhile. The further testing which I plan to carry out is thus called for, but it appears that algorithms similar to (3.5.1,11,12) do hold promise.

4. Application to Constrained Minimization

In this section we outline how the approaches considered in §2 and §3 may be applied to the constrained minimization problem.

Let $\emptyset, g_j \in C^2(\mathbb{R}^n, \mathbb{R})$, $1 \le j \le m$, and m_0 with $0 \le m_0 \le \min\{m, n-1\}$ be given. We wish to find a local minimizer of $\emptyset$ subject to the constraints $g_1 = \ldots = g_{m_0} = 0$ and $g_j \ge 0$ for $m_0 < j \le m$, i.e. if

(1) $$S = \{x \in \mathbb{R}^n \mid g_j(x) = 0,\ 1 \le j \le m_0, \text{ and } g_j(x) \ge 0,\ m_0 < j \le m\}$$

then we seek a point $x^* \in S$ such that

(2) $$\emptyset(x^*) = \min \emptyset(S \cap B(x^*, \rho))$$

for some $\rho > 0$. While we would like to do this with $\rho = +\infty$, i.e. find a global constrained minimizer x^*, we will generally have to settle for a local minimizer, just as in the unconstrained case.

Corresponding to the (unconstrained) nonsingular minimizer x^* which concerned us in §3 is the nonsingular constrained minimizer defined by

(3) Definition: A point $x^* \in S$ which satisfies (2) is a nonsingular constrained minimizer (of $\emptyset$ w.r.t. S) if $\{\nabla g_j(x^*) \mid g_j(x^*) = 0,\ 1 \le j \le m\}$ is a set of linearly independent vectors and there exist $a_j \in \mathbb{R}$, $1 \le j \le m$, such that

(4a) $$\nabla\emptyset(x^*) = \sum_{j=1}^{m} a_j \nabla g_j(x^*)$$

with

(4b) $a_j = 0$ if $g_j(x^*) > 0$ and $a_j > 0$ if $g_j(x^*) = 0$, $m_0 < j \le m$,

and also

(4c) $$p^T \left[\phi''(x^*) - \sum_{j=1}^{m} a_j g_j''(x^*) \right] p > 0$$

for all $p \neq 0$ such that $p^T \nabla g_j(x^*) = 0$ for all j with $g_j(x^*) = 0$, $1 \le j \le m$.

These conditions assure that x^* is an isolated local constrained minimizer (Fiacco and McCormick, 1968).

The a_j in (4) are Lagrange multipliers. Condition (4b) is that of strict complementary slackness. If this is relaxed to $a_j \ge 0$ for $m_0 < j \le m$ then (4a,b) become the Kuhn-Tucker conditions. These latter conditions hold at any local constrained minimizer where an appropriate constraint qualification is satisfied (see (Fiacco and McCormick, 1968) or (Mangasarian, 1969)). One such appropriate condition is the following (apparently new)

(5) Constraint Qualification: With S given by (1), $x \in S$ satisfies (5) if there exists $p \in \mathbb{R}^n$ such that

(5a) $p^T \nabla g_j(x) = 0$, $1 \le j \le m_0$

(5b) $p^T \nabla g_j(x) \ge 0$ if $g_j(x) = 0$, $m_0 < j \le m$,

(5c) $p^T \nabla g_j(x) = 0$ with $g_j(x) = 0$ for at most $n-1$ values of j and

(5d) the elements of $\{\nabla g_j(x) \mid p^T \nabla g_j(x) = 0$ and $g_j(x) = 0$, $1 \le j \le m\}$ are linearly independent.

We shall say that S satisfies (5) if each $x \in S$ satisfies (5).

If a non-Kuhn-Tucker point $x \in S$ satisfies (5), then we may state a constructive algorithm for finding a feasible direction p and thus points $y \in S$ with $\emptyset(y) < \emptyset(x)$ and $\|y-x\|$ arbitrarily small.

One way to extend the minimization algorithms proposed in §3 to algorithms for the constrained minimization problem is by means of the generalized reduced gradient method (Abadie and Carpentier, 1969). During the i^{th} major iteration this method numerically approximates a mapping $X \in C^2(\mathbb{R}^{(n-m_0)}, \mathbb{R}^n)$, which in the nondegenerate case when the number ν of $j > m_0$ such that $g_j(x_{i-1}) = 0$, say $g_{\ell(j)}(x_{i-1}) = 0$ for $1 \leq j \leq \nu$, is at most $n-m_0$, has the following properties: $X(0) = x_{i-1}$, $g_{\ell(j)}(X(z)) = e_j^T z$ for small $\|z\|$, $1 \leq j \leq \nu$, and $X(z) \in S$, whenever $\|z\|$ is small and $e_j^T z \geq 0$ for $1 \leq j \leq \nu$. Thus we may define $\bar{\emptyset} = \emptyset \cdot X \in C^2(\mathbb{R}^{(n-m_0)}, \mathbb{R})$ and consider applying one major iteration of the methods from §3 to $\bar{\emptyset}$ to obtain $x_i = X(y_{n-m_0})$ from $y_0 = 0$. The one difficulty here comes from the inequality constraints $e_j^T y_k \geq 0$, $1 \leq j \leq \nu$. However, we may handle these as follows: if $\bar{\emptyset}'(0)e_j < 0$ for some $j < \nu$ then take an Armijo-type steepest descent step in direction e_j; otherwise restrict $\bar{\emptyset}$ to $0^\nu \times \mathbb{R}^{(n-m_0-\nu)}$ and apply one major iteration of a method from §3 to this restriction, unless $\nu = n - m_0$, in which case x_{i-1} is a Kuhn-Tucker point. Near a nonsingular constrained minimizer x^* the steepest descent step will never be necessary.

A finite difference form of the reduced gradient method needs to compute X to "triple-precision" accuracy so that $\bar{\emptyset}$ may be accurately enough calculated. Note that the arithmetic precision required depends on

$$\min\left\{\left\|\nabla\phi(x_{i-1}) - \sum_{j=1}^{m} a_j \nabla g_j(x_{i-1})\right\| : a_j g_j(x_{i-1}) = 0 \text{ and } a_j \geq 0,\ m_0 < j \leq n\right\}$$

rather than $\|\nabla\phi(x_{i-1})\|$.

If S satisfies Constraint Qualification (5) and a lexicographic rule from linear programming is incorporated into the above scheme, then we can prove that any accumulation point of the major iterates is a Kuhn-Tucker point; moreover, if the iterates converge to a nonsingular constrained minimizer, if both ϕ'' and g_j'' $(i \leq j \leq m)$ satisfy a Lipschitz condition of order $\mu \in (0,1]$ near x^*, and if the calculations are done with adequate accuracy, then $O_Q(x_0, x_1, \ldots) \geq 1 + \mu$.

Another approach, which appears less cumbersome than the one above but which may involve roughly the same number of function evaluations, is to run Algorithm (2.1.54) with $\underline{f} \in C^1(\mathbb{R}^{m+n})$ given by

(6) $$\underline{f}(x,a) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_{m_0}(x) \\ a_{m_0+1} g_{m_0+1}(x) \\ \vdots \\ a_m g_m(x) \\ \nabla\phi(x) + \sum_{j=1}^{m} a_j \nabla g_j(x) \end{bmatrix}$$

If (4a,b) are satisfied, then $\underline{f}(x^*,a) = 0$. McCormick (1971) has noted that $\underline{f}'(x^*,a)$ is nonsingular if x^* is a nonsingular constrained minimizer with $a = (a_1, \ldots, a_m)^T$ as in (4a,b).

As in the unconstrained case, to assure good global convergence properties we are well advised to devise descent methods. Unfortunately, this means that the major iterates x_i must be feasible, i.e. $x_i \in S$, for otherwise it could happen that $\emptyset(x_i) < \emptyset(x^*)$ with $x_i \notin S$, in which case a descent requirement of the form $\emptyset(x_{i+1}) \leq \emptyset(x_i)$ would eventually cause breakdown of the algorithm, assuming momentarily that x^* is a global constrained minimizer. It would cost considerable effort to alter each minor iterate y_k to make it feasible. Therefore we should probably modify Algorithm (2.1.54) to take damped steps with a fairly small damping factor (see §2.6) to compute y_{m+n}, then do a search in direction $y_{m+n}-y_0$ from x_{i-1} to obtain x_i. The search would proceed, starting from x_{i-1}, by repeatedly taking a short step in direction $y_{m+n}-y_0$, then projecting the resulting point into S and checking the value of $\emptyset$ until either the composite step taken is as long as an undamped step would have been or $\emptyset$ fails to decrease sufficiently or a new g_j vanishes (becomes active).

Suppose $g_1,\ldots,g_{m_2}$ are currently active. A good way to project into S a point x which lies near S would probably be to repeatedly execute the first m_2 minor iterations of Brent's method "T1," i.e. Algorithm (2.4.1), until $\|(g_1,\ldots,g_{m_2})\|_\infty$ is suitably small. If $g_1,\ldots,g_{m_2}$ are linear, one major iteration of this process selects the point closest to x at which $g_1,\ldots,g_{m_2}$ all vanish (assuming the gradients of $g_1,\ldots,g_{m_2}$ to be linearly independent, with distance measured by the 2-norm $\|\cdot\|$).

When Algorithm (2.1.54) is run with $\underline{f}$ given by (6), advantage may be taken of linear constraints--see §2.6. With any linear constraints ordered first (and followed by any nearly linear constraints), it is probably best to choose $q_j = e_j$ for $1 \leq j \leq m$. Assume $g_j(x_{i-1}) = 0$ for $1 \leq j \leq m_1 \leq n-1$ (with $m_1 \geq m_0$) and $g_j(x_{i-1}) > 0$ for $m_1 < j \leq m$: for simplicity we shall ignore the dependence of the major iterates on their last m components $a_1,\dots,a_m$ and refer to these separately; similarly, let y_k denote the first n components only of the k^{th} minor iterate. The first m_1 minor iterations result in w_j of the form $w_j = \begin{pmatrix}\underline{w}_j\\ 0\end{pmatrix}$, $1 \le j \le m_1$, where $\underline{w}_j$ would have been obtained if $\underline{f}$ mapped $\mathbb{R}^n \to \mathbb{R}^n$ and had $g_1,\dots,g_{m_1}$ for its first m_1 components. (Note that if x_{i-1} is feasible, then the first m_1 minor iterations serve only to produce $\underline{w}_1,\dots,\underline{w}_{m_1}$.) The next $m-m_1$ minor iterations result in $a_j = 0$, $m_1 < j \le m$, with $w_j = \begin{pmatrix}0_n\\ e_j\end{pmatrix} = e_{n+j}$, and need not be explicitly performed. To choose $q_{m+1},\dots,q_{m+n-m_1}$, let $\hat{q}_1,\dots,\hat{q}_{m_1}$ be an orthonormal basis for $\mathrm{Span}\{\underline{w}_1,\dots,\underline{w}_{m_1}\}$ and for $m < k \le m+n-m_1$ choose

$$p_k = \begin{pmatrix}\hat{p}_k\\ 0_m\end{pmatrix} \in \mathrm{Span}\{w_1,\dots,w_{k-1}\}^{\perp} = \mathrm{Span}\{u_k^{(k)},\dots,u_n^{(k)}\} \text{ with } \hat{p}_k \in \mathbb{R}^n, \text{ let}$$

$\tilde{\hat{q}}_k = (I-\hat{Q}_{k-1}\hat{Q}_{k-1}^T)\hat{p}_k$, where $\hat{Q}_{k-1}=[\hat{q}_1,\dots,\hat{q}_{m_1},\hat{q}_{m+1},\dots,\hat{q}_{k-1}]$, let $\hat{q}_k=\tilde{\hat{q}}_k/\|\tilde{\hat{q}}_k\|$, and $q_k = \begin{pmatrix}0_m\\ \hat{q}_k\end{pmatrix}$. This is analogous to the choice made in Algorithm (3.3.8). Therefore the same descent tests will be satisfied near a nonsingular constrained minimizer and hence may be applied to forestall wasted computational effort, as in (3.3.8). The remaining q_k, $m+n-m_1 < k \le m+n$, may be chosen as $q_k = \begin{pmatrix}0_m\\ \hat{q}_{k-(m+n-m_1)}\end{pmatrix}$, which results in the Lagrange multipliers $a_1,\dots,a_{m_1}$ being computed. At the end of the current major iteration (or start of the next) $\|\nabla\phi(x_i) - \sum_{j=1}^{m} a_j\nabla g_j(x_i)\|$ may thus be approximated for use in determining the arithmetic precision needed or in testing for convergence.

We may imbed the above in an algorithm similar to (3.4.9), which proceeds with the first $m+1$ minor iterations, then uses tests like (3.4.9k,n) to decide whether to proceed as above or simply take an Armijo step and then start the next major iteration. Thus we may design an algorithm like (3.4.9) such that if S satisfies Constraint Qualification (5), then any accumulation point of the major iterates is a Kuhn-Tucker point and if a nonsingular constrained minimizer x^* is an accumulation point of the iterates, if ϕ and the g_j satisfy a Lipschitz condition of order $\mu \in (0,1]$ near x^*, and if the control constants are properly chosen, then the major iterates converge to x^* with Q-order $\mathcal{O}_Q(x_0, x_1, \ldots) \geq 1 + \mu$.

5. Conclusion

Newton's method attempts to find a zero x^* of $f \in C^1(\mathbb{R}^n)$ by taking a step intended to make all components of f vanish at once. In this respect Newton's method processes the components of f in parallel. On the other hand, Brown's method and the generalizations thereof considered here process the components of f serially, one at a time. One major iteration of these methods may be described as follows: given a starting point y_0, linearize the first component f_1 of f at y_0; this linearization vanishes in a certain hyperplane H_1; find a point $y_1 \in H_1$ and linearize the restriction of f_2 to H_1 at y_1; this linearization vanishes in an n-2 dimensional hyperplane H_2; find a point $y_2 \in H_2$; in general, having found an n-k dimensional hyperplane H_k in which linearizations of $f_1,\ldots,f_k$ vanish, find $y_k \in H_k$, restrict f_{k+1} to H_k, linearize this restriction, and find the hyperplane H_{k+1} in which this linearization vanishes; stop when y_n has been found: y_n is the unique point in $\mathbb{R}^n$ at which all n linearizations vanish; let y_n be the next major iterate. When f is a general nonlinear function and finite differences are used in constructing the linearizations, this approach has the advantage of having to do the work of determining only about half the components of f' and thus requires only about half as many function evaluations per iteration as the corresponding finite difference Newton's method, while enjoying the same rate of (local) convergence.

We have examined Brown's method from a somewhat different point of view than the one Brown had when he proposed his method. Our view, which was suggested in a paper by Brent (1973), led us to a more efficient implementation of Brown's method and let us see the connaction between his

The extensions sketched in §4 of our ideas to the constrained minimization problem also seem to hold promise. They surely merit further development and testing.

method, Brent's variation thereof, and Craig's method for solving linear equations. It led us further to propose new descent algorithms for the unconstrained and constrained minimization problems. One of these algorithms is closely related to some generalized conjugate gradient algorithms and requires about half as many function evaluations per major iteration (reinitialization plus n steps) as what may be the most reasonable corresponding finite difference version of these conjugate gradient methods, while retaining the same rate of convergence, albeit with increased housekeeping overhead. (This of course says nothing about nonlocal behavior.)

When the finite difference Newton's method is applied to the unconstrained minimization problem with $f = \nabla\phi$, where $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ is the function to be minimized, advantage may be taken of the symmetry of $f' = \phi''$, thus reducing to $n^2/2 + \mathcal{O}(n)$ the number of function evaluations required per (major) iteration. Locally, at least, our approach thus enjoys no advantage in terms of function evaluations over the finite difference Newton's method. Indeed, the test program described in §3.6 requires $n^2 + \mathcal{O}(n)$ function evaluations per major iteration. However, as noted in §3.6, for some of the proposed algorithms this number may be cut in half. Moreover, since our approach processes the rotated components of $\nabla\phi$ serially, we may terminate a major iteration early if the effort required to complete it does not appear warranted. Under some conditions our approach might thus require less effort in terms of function evaluations than the finite difference Newton's method.

More experimentation is needed to determine the value of the approach to unconstrained minimization proposed in §3. In particular, we have suggested several modifications which deserve testing. At this point our approach does appear to hold promise of practical value.

Faddeev, D. K., and Faddeeva, V. N. (1963), Computational Methods of Linear Algebra, W. H. Freeman, London.

Fiacco, A. V., and McCormick, G. P. (1968), Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York.

Fletcher, R., and Reeves, C. M. (1964), "Function Minimization by Conjugate Gradients," Computer Journal 7, pp. 149-154.

Forsythe, G., and Moler, C. B. (1967), Computer Solution of Linear Algebraic Systems, Prentice Hall, Englewood Cliffs, N. J.

Golub, G., and Kahan, W. (1965), "Calculating the Singular Values and Pseudo-inverse of a Matrix," SIAM J. Num. Anal. 2, No. 2, pp. 205-224.

Mangasarian, O. L. (1969), Nonlinear Programming, McGraw-Hill, New York.

McCormick, G. P. (1971), "Penalty Function Versus Nonpenalty Function Methods for Constrained Nonlinear Programming Problems," Math. Prog. 1, pp. 217-238.

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

Ostrowski, A. M. (1960), Solution of Equations and Systems of Equations, Academic Press, New York.

Paige, C. C. (1974), "Bidiagonalization of Matrices and Solution of Linear Equations," SIAM J. Numer. Anal. 11, No. 1, pp. 197-209.

Polak, E., and Ribière, G. (1969), "Note sur la Convergence de Methodes de Directions Conjugees", Rev. Française Informat. Recherche Opérationelle 16-R1, pp. 35-43.

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

Rosenbrock, H. H. (1960), "An Automatic Method for Finding the Greatest or the Least Value of a Function," Computer Journal 3, pp. 175-184.

Shamanskii, V. E. (1967), "A Modification of Newton's Method," Ukrain. Mat. Zh. 19, pp. 133-138 (in Russian).

Stewart, G. W. (1972), "Conjugate Direction Algorithms and Matrix Decompositions," University of Texas at Austin, Center for Numerical Analysis, Technical Report CNA-49 (July 1972).

6. References

Abadie, J., and Carpentier, J. (1969), "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in Optimization, edited by R. Fletcher, pp. 37-49, Academic Press, London.

Armijo, L. (1966), "Minimization of Functions Having Lipschitz Continuous First Partial Derivatives," Pacific Journal of Math. 16, No. 1, pp. 1-3.

Brent, R. P. (1973), "Some Efficient Algorithms for Solving Systems of Nonlinear Equations," SIAM J. Numer. Anal. 10, No. 2, pp. 327-344.

Brown, K. M. (1966), "A Quadratically Convergent Method for Solving Simultaneous Nonlinear Equations," Ph.D. Thesis, Purdue University, Department of Computer Sciences, August 1966.

Brown, K. M. (1973), "Computer Oriented Algorithms for Solving Systems of Simultaneous Nonlinear Algebraic Equations," in Numerical Solution of Systems of Nonlinear Algebraic Equations, edited by G. D. Byrne and C. A. Hall, Academic Press, New York.

Brown, K. M., and Dennis, J. E., Jr. (1971), "On the Second Order Convergence of Brown's Derivative-free Method for Solving Simultaneous Nonlinear Equations," Yale University Department of Computer Science, Research Report No. 71-7.

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

Burmeister, W. (1973), "Die Konvergenzordnung des Fletcher-Powell-Algorithmus," ZAMM 53, pp. 693-699.

Cohen, A. I. (1972), "Rate of Convergence of Several Conjugate Gradient Algorithms," SIAM J. Num. Anal. 9, pp. 248-259.

Craig, E. J. (1955), "The N-Step Iteration Procedures," J. of Math. and Physics 34, pp. 64-73.

Daniel, J. W. (1967), "The Conjugate Gradient Method for Linear and Nonlinear Operator Equations," SIAM J. Num. Anal. 4, pp. 10-26.

Dennis, J. E., Jr. (1970), "On the Convergence of Newton-like Methods," in Numerical Methods for Nonlinear Algebraic Equations, edited by P. Rabinowitz, Gordon-Breach, London.

Dixon, L. C. W. (1972), "Variable Metric Algorithms: Necessary and Sufficient Conditions for Identical Behavior on Non-quadratic Functions," J.O.T.A. 10, pp. 34-40.

Dixon, L. C. W. (1973), "Conjugate Directions without Linear Searches," J. Inst. Maths. Applics. 11, pp. 317-328.

$[s,t]$	$\{r \in \mathbb{R} \cup \{-\infty,\infty\} \mid s \leq r \leq t\}$ for $s,t \in \mathbb{R} \cup \{-\infty,\infty\} = [-\infty,\infty]$
$[s,t)$	$\{r \in [-\infty,\infty] \mid s \leq r < t\}$ for $s,t \in [-\infty,\infty]$
$(s,t]$	$\{r \in [-\infty,\infty] \mid s < r \leq t\}$ for $s,t \in [-\infty,\infty]$
(s,t)	$\{r \in \mathbb{R} \mid s < r < t\}$ for $s,t \in [-\infty,\infty]$
$B(x,\rho)$	$\{y \in \mathbb{R}^n : \lVert y-x \rVert < \rho\}$ for $x \in \mathbb{R}^n$, $\rho \in (0,\infty]$
$\bar{B}(x,\rho)$	$\{y \in \mathbb{R}^n : \lVert y-x \rVert \leq \rho\}$ for $x \in \mathbb{R}^n$, $\rho \in (0,\infty]$
$\mathbb{N}$	$\{0,1,2,\ldots\}$, the set of natural numbers
$\mathbb{N}^+$	$\{1,2,\ldots\}$, the set of positive integers
$C^k(\mathbb{R}^n, \mathbb{R}^m)$	the set of all k-times continuously Frechet differentiable (continuous) mappings from $\mathbb{R}^n$ to $\mathbb{R}^m$, $k \in \mathbb{N}$
$C^k(\mathbb{R}^n)$	$C^k(\mathbb{R}^n, \mathbb{R}^n)$
g'	the Frechet derivative of $g \in C^1(\mathbb{R}^n, \mathbb{R}^m)$, i.e. the Jacobian matrix of g
$\nabla\phi$	$(\phi')^T$ for $\phi \in C^1(\mathbb{R}^u, \mathbb{R})$
ϕ''	the Hessian of $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$
$\frac{\partial\phi}{\partial u}(x)$	$\phi'(x)u$, where $\phi \in C^1(\mathbb{R}^n, \mathbb{R})$ and $x, u \in \mathbb{R}^n$
$\frac{\partial^2\phi}{\partial u \partial v}(x)$	$v^T\phi''(x)u$, where $\phi \in C^2(\mathbb{R}^n, \mathbb{R})$ and $x,u,v \in \mathbb{R}^n$
$\frac{\partial^2\phi}{\partial u^2}$	$\frac{\partial^2\phi}{\partial u \partial u}$
$\frac{\Delta\phi}{\Delta u}(x,h)$	$\frac{\phi(x+hy)-\phi(x)}{h}$ (see (1.2.1,13))
$\frac{\tilde{\Delta}\phi}{\Delta u}(x,h)$	$\frac{\phi(x+hu)-\phi(x-hu)}{2h}$ (see (1.2.3,16))

7. Appendix: Notation

Symbol	Meaning
ϵ	"is an element of"
$\forall$	"for all"
$\cdot$	composition: $f \cdot g(x) \equiv f(g(x))$
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^u$	Euclidean n-space (usual topology)
$\mathcal{L}(\mathbb{R}^u, \mathbb{R}^m)$	the set of real $m \times n$ matrices
$\mathcal{L}(\mathbb{R}^n)$	$\mathcal{L}(\mathbb{R}^n, \mathbb{R}^n)$
x^T	the transpose of $x \in \mathbb{R}^n$ (x is a column vector, i.e. $n \times 1$ matrix, while x^T is a row vector, i.e. $1 \times n$ matrix)
A^T	the transpose of matrix A
e_j	the j^{th} standard unit vector (j^{th} component equals 1, remaining components, whose number depends on context, equal 0).
$\lVert x \rVert_p$	$\left[\sum_{j=1}^{n} (e_j^T x)^p\right]^{1/p}$ if $x \in \mathbb{R}^n$ and $1 \leq p < +\infty$
$\lVert x \rVert_\infty$	$\max\{\lvert e_j^T x\rvert : 1 \leq j \leq n\}$ if $x \in \mathbb{R}^n$
$\lVert x \rVert$	$\lVert x \rVert_2$
$\lVert A \rVert_p$	$\max\{\lVert Ax \rVert_p : \lVert x \rVert_p = 1, x \in \mathbb{R}^n\}$ if $a \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ and $1 \leq p \leq \infty$
$[v_1, \ldots, v_k]$	the $n \times k$ matrix whose j^{th} column is $v_j \in \mathbb{R}^n$
I_k	$[e_1, \ldots, e_k]$, the $k \times k$ identity matrix
I	I_k, where k is clear from context

$\frac{\Delta^2\phi}{\Delta u \Delta v}(x,h)$	$\frac{\phi(x+h[u+v]) - \phi(x+hu) - \phi(x+hv) + \phi(x)}{h^2}$ (see (1.2.5,18))
$\frac{\Delta^2\phi}{\Delta u^2}$	$\frac{\Delta^2\phi}{\Delta u \Delta u}$
$(f-g)(x)$	$f(x) - g(x)$, where $f,g\colon \mathbb{R}^n \to \mathbb{R}^m$ and $x \in \mathbb{R}^n$
$g(S)$	$\{g(s) \mid s \in S\}$, where $g\colon X \to Y$ and $S \subset X$
$\chi(A)$	the ratio of the largest to the smallest singular value of $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$--see §1.2; $\chi(A) = +\infty$ if A does not have full rank
A^+	the pseudoinverse of $A \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$--see §1.2
$\mathrm{diag}(d_1,\dots,d_m)$	$m \times m$ diagonal matrix whose $k^{\underline{th}}$ diagonal element is d_k
§2.0	the introduction to section 2
RHS(8)	the right-hand side of formula (8) of the current subsection
LHS(2.4.5)	the left-hand side of formula (5) of §2.4
WLOG	without loss of generality
MVT	the mean value theorem
$\ell n(t)$	the natural logarithm of t
w.r.t.	with respect to