is contradicted if it is assumed that
m (2.5.3) and (2.5.8) that

(2.5.12)

1 of Lipschitz continuity of $\mathbf{g}$ is made
: level set) then it follows from (2.5.9)
rating this result with (2.5.12) gives

^{0}. (2.5.13)

f is bounded below, it follows that
tion is being sought, it is assumed
erty that

(2.5.14)

$\|\mathbf{g}^{(k)}\| = 0$ follows. It can be seen that
os $\theta^{(k)} \geqslant \mu > 0$, and to my knowledge
used to prove global convergence
iods satisfy (2.5.14) have been given
Baali (1985).
ests with the property of superlinear
. Using the notation $\mathbf{h}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$
ns, then the following result can be

is positive definite, and if $\mathbf{x}^{(k)} \to \mathbf{x}^*$

0.

h Taylor series, continuity and the
, the following expressions can be

$^{2})$

where $\lambda > 0$ is the least eigenvalue of $\mathbf{G}^*$. The lemma follows directly from these expressions. □

It follows from the lemma that tests such as (2.5.3) and (2.5.5) are satisfied for all k sufficiently large under these conditions. Consider therefore any line search algorithm which is locally superlinearly convergent when $\alpha^{(k)} = 1$ for all k. If the initial estimate of the step in the line search algorithm is taken asymptotically as $\alpha_1 = 1$ then it follows ultimately that this value is accepted as $\alpha^{(k)}$. Hence $\alpha^{(k)} = 1$ for all k sufficiently large, and the superlinear convergence property is preserved. This result is true for algorithms based on any of the tests (2.5.1), (2.5.2), (2.5.4) and (2.5.6).

2.6 ALGORITHMS FOR THE LINE SEARCH SUBPROBLEM

Numerous line search algorithms have been proposed over the years, and a good choice is important since it can have a considerable effect on the performance of the method in which it is embedded. The availability or not of first derivatives is a primary consideration: if derivatives are not available then there is not much theory available to act as a guide to how the line search should be terminated. There are therefore a wide range of possibilities and some of these are considered towards the end of this section. However, the case in which first derivatives are available is of major importance. In this case it is widely accepted that the line search should attempt to satisfy the conditions (2.5.1) and (2.5.6) for the reasons set out in Section 2.5. Therefore this section describes a line search algorithm which satisfies these conditions by making a finite (usually small) number of evaluations of $f(\alpha)$ and $f'(\alpha)$. (The notation $f(\alpha) = f(\mathbf{x}^{(k)} + \alpha\mathbf{s}^{(k)})$ as in Section 2.5 is used and the descent condition $f'(0) < 0$ is assumed to hold.)

A line search algorithm is an iterative method which generates a sequence of estimates $\{\alpha_i\}$. The sequence terminates when an iterate is located which satisfies some standard conditions for an acceptable point. There are two distinct parts to any line search algorithm. First comes the *bracketing phase* which searches to find a *bracket*, that is a non-trivial interval, $[a_i, b_i]$ say, which is known to contain an interval of acceptable points. This is followed by the *sectioning phase* in which the bracket is *sectioned* (i.e. divided) so as to generate a sequence of brackets $[a_j, b_j]$ whose length tends to zero. This forms the basis of a finite termination proof for the line search algorithm. In addition, since it is preferable to find an acceptable point which is close to a local minimizer of $f(\alpha)$, some form of *interpolation* is also desirable. This involves fitting usually a quadratic or cubic polynomial in α to known data, and choosing the next iterate α_{j+1} so as to minimize the polynomial, possibly subject to some sectioning restrictions.

One particular line search algorithm is now described for the case that first derivatives are available. Whilst it is not necessarily uniquely best, it is the one which I currently prefer for general unconstrained minimization, and it is closely similar to other line search algorithms that are also currently used. Some results which substantiate the properties of the algorithm are also proved. It has already been pointed out in Lemma 2.5.2 *either* that there exist acceptable points, *or* that the graph of $f(\alpha)$ never intersects the ρ-line. The latter possibility is avoided by assuming that the user is able to supply a lower bound $\bar{f}$ on $f(\alpha)$. (More precisely it is assumed that the user is prepared to accept any value of $f(\alpha)$ for which $f(\alpha) \leqslant \bar{f}$.) For example in a nonlinear least squares problem $\bar{f}=0$ would be appropriate. A consequence of this assumption is that the line search can be restricted to the interval $(0, \mu]$ where

$$\mu = (\bar{f} - f(0))/(\rho f'(0)) \tag{2.6.1}$$

is the point at which the ρ-line intersects the line $f = \bar{f}$.

In the bracketing phase the iterates α_i move out to the right in increasingly large jumps until either $f \leqslant \bar{f}$ is detected or a bracket on an interval of acceptable points is located. Initially $\alpha_0 = 0$, α_1 is given $(0 < \alpha_1 \leqslant \mu)$, and this phase of the algorithm can be described as follows:

<u>for</u> $i := 1, 2, \ldots$ <u>do</u>

<u>begin</u> evaluate $f(\alpha_i)$;

<u>if</u> $f(\alpha_i) \leqslant \bar{f}$ <u>then</u> terminate; ρ

<u>if</u> $f(\alpha_i) > f(0) + \alpha_i f'(0)$ <u>or</u> $f(\alpha_i) \geqslant f(\alpha_{i-1})$

<u>then begin</u> $a_i := \alpha_{i-1}$; $b_i := \alpha_i$; terminate B <u>end</u>;

evaluate $f'(\alpha_i)$; (2.6.2)

<u>if</u> $|f'(\alpha_i)| \leqslant -\sigma f'(0)$ <u>then</u> terminate;

<u>if</u> $f'(\alpha_i) \geqslant 0$

<u>then begin</u> $a_i := \alpha_i$; $b_i := \alpha_{i-1}$; terminate B <u>end</u>;

<u>if</u> $\mu \leqslant 2\alpha_i - \alpha_{i-1}$

<u>then</u> $\alpha_{i+1} := \mu$

<u>else</u> choose $\alpha_{i+1} \in [2\alpha_i - \alpha_{i-1}, \min(\mu, \alpha_i + \tau_1(\alpha_i - \alpha_{i-1}))]$

<u>end</u>.

In this algorithm $\tau_1 > 1$ is a preset factor by which the size of the jumps is increased, typically $\tau_1 = 9$. The choice of α_{i+1} in the next last line of (2.6.2) can be made in any way, but a sensible choice would be to minimize in the given interval a cubic polynomial interpolating $f(\alpha_i)$, $f'(\alpha_i)$, $f(\alpha_{i-1})$ and $f'(\alpha_{i-1})$. Because μ exists and by virtue of (2.6.1), algorithm (2.6.2) clearly must terminate. In (2.6.2) 'terminate B' is used to indicate termination of just the bracketing phase, whereas 'terminate' alone indicates termination of the line search with an acceptable point α_i, or a point for which $f(\alpha_i) \leqslant \bar{f}$. If 'terminate B' occurs then a bracket $[a_i, b_i]$ is known (it is convenient to allow either $a_i < b_i$ or $b_i < a_i$ in this notation for a bracket) which satisfies the following properties.

(i) a_i is the current best trial point (least f) that also satisfies (2.5.1).
(ii) $f'(a_i)$ has been evaluated and satisfies

$$(b_i - a_i) f'(a_i) < 0 \text{ but not (2.5.6).} \tag{2.6.3}$$

(iii) b_i satisfies either $f(b_i) > f(0) + b_i \rho f'(0)$ or $f(b_i) \geqslant f(a_i)$ or both.

For such a bracket the following result holds.

Lemma 2.6.1

If $\sigma \geqslant \rho$, a bracket which satisfies (2.6.3) *contains an interval of acceptable points for* (2.5.1) *and* (2.5.6).

Proof

If $b_i > a_i$, consider a line, L say, through $(a_i, f(a_i))$ having slope $\rho f'(0)$ (i.e. parallel to the ρ-line). Let $\hat{\alpha}_i$ be the point in (a_i, b_i) closest to a_i at which the graph of $f(\alpha)$ intersects L. Existence of $\hat{\alpha}_i$ comes from (ii) and (iii) and continuity. Use of the mean value theorem as in Lemma 2.5.1 then implies the required result. If $b_i < a_i$ then a line through $(a_i, f(a_i))$ having zero slope is considered and a similar argument is used. □

This lemma shows that the bracketing phase has achieved its aim of bracketing an interval of acceptable points. Next comes the sectioning phase which generates a sequence of brackets $[a_j, b_j]$ for $j = i, i+1, \ldots$ whose lengths tend to zero. Each iteration picks a new trial point α_j in $[a_j, b_j]$ and the next bracket is either $[a_j, \alpha_j]$, $[\alpha_j, a_j]$ or $[\alpha_j, b_j]$, the choice being made so that properties (2.6.3) are preserved. The sectioning phase terminates when the current trial point α_j is found to be acceptable in (2.5.1) and (2.5.6). The algorithm can be described as follows

<u>for</u> $j := i, i+1, \ldots$ <u>do</u>
<u>begin</u> choose $\alpha_j \in [a_j + \tau_2(b_j - a_j),\ b_j - \tau_3(b_j - a_j)]$;
evaluate $f(\alpha_j)$;
<u>if</u> $f(\alpha_j) > f(0) + \rho\alpha_j f'(0)$ <u>or</u> $f(\alpha_j) \geqslant f(a_j)$
<u>then</u> <u>begin</u> $a_{j+1} := a_j$; $b_{j+1} := \alpha_j$ <u>end</u>
<u>else</u> <u>begin</u> evaluate $f'(\alpha_j)$; (2.6.4)
<u>if</u> $|f'(\alpha_j)| \leqslant -\sigma f'(0)$ <u>then</u> terminate;
$a_{j+1} := \alpha_j$;
<u>if</u> $(b_j - a_j) f'(\alpha_j) \geqslant 0$ <u>then</u> $b_{j+1} := a_j$ <u>else</u> $b_{j+1} := b_j$
<u>end</u>.
<u>end</u>.

In this algorithm τ_2 and τ_3 are preset factors $(0 < \tau_2 < \tau_3 \leqslant \frac{1}{2})$ which restrict α_j from being arbitrarily close to the extremes of the interval $[a_j, b_j]$. It then follows that

$$|b_{j+1} - a_{j+1}| \leqslant (1 - \tau_2)|b_j - a_j| \tag{2.6.5}$$

and this guarantees convergence of the interval lengths to zero. Typical values are $\tau_2 = \frac{1}{10}$ ($\tau_2 \leqslant \sigma$ is advisable) and $\tau_3 = \frac{1}{2}$, although the algorithm is insensitive to the precise values that are used. The choice of α_j in the second line of (2.6.4) can be made in any way, but a sensible choice would be to minimize in the given interval a quadratic or cubic polynomial which interpolates $f(a_j)$, $f'(a_j)$, $f(b_j)$, and $f'(b_j)$ if it is known. When (2.6.4) terminates then α_j is the required acceptable point and becomes the step length $\alpha^{(k)}$ to be used in (2.3.2). The convergence properties of the sectioning scheme are given in the following result (Al-Baali and Fletcher, 1986).

Theorem 2.6.1

If $\sigma \geqslant \rho$ and the initial bracket $[a_i, b_i]$ satisfies (2.6.3), then the sectioning scheme (2.6.4) has the following properties.

Either (a) *the scheme terminates with an α_j which is an acceptable point in (2.5.1) and (2.5.6)*

or (b) *the scheme fails to terminate and there exists a point c such that for j sufficiently large $a_j \uparrow c$ and $b_j \downarrow c$ monotonically (not strictly) and c is an acceptable point in (2.5.1) and (2.5.6).*

Moreover if $\sigma > \rho$ then (b) cannot occur and the scheme must terminate.

Proof

It follows from (2.6.5) that $|a_j - b_j| \to 0$ and from (2.6.4) that $[a_{j+1}, b_{j+1}] \subset [a_j, b_j]$. Therefore $\exists$ a limit point c such that $a_j \to c$, $b_j \to c$ and hence $\alpha_j \to c$. Since a_j satisfies (2.5.1) but not (2.5.6) it follows that c satisfies (2.5.1) and that

$$|f'(c)| \geqslant -\sigma f'(0) \tag{2.6.6}$$

Let there exist an infinite subsequence of brackets for which $b_j < a_j$. It follows from (2.6.3) (iii) that $f(b_j) - f(a_j) \geqslant 0$ and hence by the mean value theorem and the existence of c that $f'(c) \leqslant 0$. But from (2.6.3) (ii), $f'(a_j)(b_j - a_j) < 0$ implies that $f'(c) \geqslant 0$ in the limit. These inequalities contradict (2.6.6), proving that $a_j \uparrow c$ and $b_j \downarrow c$ for j sufficiently large. Consider therefore the case that $b_j > a_j$. It follows from (2.6.3) (iii) and a_j satisfying (2.5.1) that

$$f(b_j) - f(a_j) \geqslant (b_j - a_j)\rho f'(0),$$

so the mean value theorem and the existence of c imply that $f'(c) \geqslant \rho f'(0)$. But $f'(a_i)(b_i - a_i) < 0$ implies that $f'(c) \leqslant 0$. If $\sigma > \rho$, these inequalities contradict (2.6.6), showing that case (b) cannot arise, in which case the algorithm must terminate as in (a). If $\sigma = \rho$ the possibility of non-termination exists, with a limit point c for which (2.5.1) holds and $f'(c) = \rho f'(0)$. □

An example in which case (b) arises is described by Al-Baali and Fletcher (1986). This reference also gives a similar theorem for more simple algorithm aimed at satisfying the Wolfe–Powell conditions. For practical purposes, however, Theorem 2.6.1 indicates that $\sigma > \rho$ should be selected, in which case the sectioning algorithm (2.6.4) is guaranteed to terminate with an acceptable point. This algorithm is not the only one which has been used, for instance a different one is given by Moré and Sorensen (1982). This differs from (2.6.4) in that α_j is allowed to become arbitrarily close to a_j, but a bisection step is used if the bracket length is not reduced by a factor of $\frac{1}{2}$ after any two iterations. This is similar to an idea of Brent (1973a) in the context of solving a nonlinear equation.

The use of polynomial extrapolation or interpolation in conjunction with (2.6.2) and (2.6.4) is recommended. Given an interval $[0, 1]$, and data values f_0, f'_0 and $f_1 (f_0 = f(0)$ etc.) then the unique quadratic that interpolates these values is defined by

$$q(z) = f_0 + f'_0 z + (f_1 - f_0 - f'_0)z^2.$$

If in addition f'_1 is given then the corresponding (Hermite) interpolating cubic is

$$c(z) = f_0 + f'_0 z + \eta z^2 + \xi z^3$$

where

$$\eta = 3(f_1 - f_0) - 2f'_0 - f'_1$$

and

$$\xi = f'_0 + f'_1 - 2(f_1 - f_0).$$

In the schemes (2.6.2) and (2.6.4) it is required to work with an interval $[a, b]$ rather than $[0, 1]$, and $a > b$ is allowed. To do this the transformation

$$\alpha = a + z(b - a)$$

is used which maps $[0, 1]$ into $[a, b]$. The chain rule gives $\mathrm{d}f/\mathrm{d}z = \mathrm{d}f/\mathrm{d}\alpha \cdot \mathrm{d}\alpha/\mathrm{d}z = (b - a)\mathrm{d}f/\mathrm{d}\alpha$ which relates the derivatives f'_0 and f'_1 above to known values of $\mathrm{d}f/\mathrm{d}\alpha$ obtained in the line search. To find the minimizers of $q(z)$ or $c(z)$ in a given interval, it is necessary to examine not only the stationary values, but also the values and derivatives at the ends of the interval. This requires some simple calculus.

An example is now given of the use of this line search, illustrating the bracketing, sectioning and interpolation processes. Consider using the function (1.2.2) and let $\mathbf{x}^{(k)} = \mathbf{0}$ and $\mathbf{s}^{(k)} = (1, 0)^T$. Then $f(\alpha) = 100\alpha^4 + (1 - \alpha)^2$ and $f'(\alpha) = 400\alpha^3 - 2(1 - \alpha)$. The parameters $\sigma = 0.1$, $\rho = 0.01$, $\tau_1 = 9$, $\tau_2 = 0.1$ and $\tau_3 = 0.5$ are used. The progress of the line search is set out in Table 2.6.1. The first part shows what happens if the initial guess is $\alpha_1 = 0.1$. The initial interval does not give a bracket, so the bracketing algorithm requires that the next iterate is chosen in $[0.2, 1]$. Mapping $[0, 0.1]$ on to $[0, 1]$ in z-space, the resulting cubic fit is $c(z) = 1 - 0.2z + 0.02z^3$ and the minimum value of $c(z)$ in $[2, 10]$ is at $z = 2$. Thus $\alpha_2 = 0.2$ is the next iterate. This iterate give a bracket $[0.2, 0.1]$ which satisfies (2.6.3). The sectioning algorithm comes

Table 2.6.1 The line search using first derivatives

Starting from $\alpha_1 = 0.1$				
α	0	0.1	0.2	0.160948
$f(\alpha)$	1	0.82	0.8	0.771111
$f'(\alpha)$	-2	-1.4	1.6	-0.010423

Starting from $\alpha_1 = 1$					
α	0	1	0.1	0.19	0.160922
$f(\alpha)$	1	100	0.82	0.786421	0.771112
$f'(\alpha)$	-2	–	-1.4	1.1236	-0.011269

into play and seeks a new iterate in $[0.19, 0.15]$. Mapping $[0.2, 0.1]$ on to $[0, 1]$ gives a cubic $c(z) = 0.8 - 0.16z + 0.24z^2 - 0.06z^3$. The minimizer of this cubic in $[0.1, 0.5]$ is locally unconstrained at $z = 0.390524$, and this gives a new iterate $\alpha_3 = 0.160948$. This point is found to be acceptable and the line search is terminated. Alternatively let the line search start with $\alpha_1 = 1$. Then the initial value of $f(1) > f(0)$ so this point immediately gives a bracket and it is not necessary to evaluate $f'(1)$ (this saves unnecessary gradient evaluations). The sectioning phase is therefore entered. Making a quadratic interpolation gives $q(\alpha) = 1 - 0.2\alpha + 99.2\alpha^2$, which is minimized in $[0.1, 0.5]$ by $\alpha_2 = 0.1$. Thus $[0.1, 1]$ is the new bracket that satisfies (2.6.3) and the next iterate is chosen as a point in $[0.19, 0.55]$. This could be done either by interpolating a quadratic at $\alpha = 0.1$ and $\alpha = 1$, or interpolating a cubic at $\alpha = 0$ and $\alpha = 0.1$ (my current algorithm would do the latter). In both cases here, $\alpha_3 = 0.19$ is the minimizing point and $[0.19, 0.1]$ becomes the new bracket. A cubic interpolation in this bracket leads finally to an iterate $\alpha_4 = 0.160922$ which is acceptable.

This description of the algorithm neglects the effect of round-off errors, and these can cause difficulties when $\mathbf{x}^{(k)}$ is close to $\mathbf{x}^*$. In this case, although $f'(0) < 0$, it may happen that $f(\alpha) \geqslant f(0)$ for all $\alpha > 0$, due to round-off error. This situation can also arise if the user has made errors in his formulae for derivatives. Therefore I have found it advisable also to terminate after line 3 of (2.6.4) if $(a_j - \alpha_j) f'(a_j) \leqslant \varepsilon$ where ε is a tolerance on f such as in (2.3.7). This causes an exit from the minimization method with an indication that no progress can be made in the line search.

The initial choice of α_1 also merits some study. If an estimate $\Delta f > 0$ is available of the likely reduction in f to be achieved by the line search, then it is possible to interpolate a quadratic to $f'(0)$ and Δf, giving the estimate

$$\alpha_1 = -2\Delta f / f'(0). \tag{2.6.7}$$

On the first iteration of the minimization method Δf must be user supplied, but subsequently $\Delta f = \max(f^{(k-1)} - f^{(k)}, 10\varepsilon)$ has been found to work well, that is the reduction from the previous iteration, suitably safeguarded. For Newton-like methods, however, the choice $\alpha^{(k)} = 1$ is significant in giving rapid