STEPLENGTH ALGORITHMS FOR MINIMIZING A CLASS OF NONDIFFERENTIABLE FUNCTIONS

by

Walter Murray and Michael L. Overton

STAN-CS-78-679
NOVEMBER 1978

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

STANFORD JUNIOR UNIVERSITY
ORGANIZED 1891

STEPLENGTH ALGORITHMS FOR MINIMIZING A CLASS OF NONDIFFERENTIABLE FUNCTIONS.

Walter Murray
Division of Numerical Analysis and Computer Science
National Physical Laboratory
Teddington, Middlesex
England TW11 OLW

and

Michael L. Overton[*]
Computer Science Department
Stanford University
Serra House, Serra Street
Stanford, California
94305

[*] The work of this author was supported in part by the National Research Council of Canada and in part by the National Science Foundation Grant MCS 75-13497-A01.

Abstract.

Four steplength algorithms are presented for minimizing a class of nondifferentiable functions which includes functions arising from ℓ_1 and ℓ_∞ approximation problems and penalty functions arising from constrained optimization problems. Two algorithms are given for the case when derivatives are available wherever they exist and two for the case when they are not available. We take the view that although a simple steplength algorithm may be all that is required to meet convergence criteria for the overall algorithm, from the point of view of efficiency it is important that the step achieve as large a reduction in the function value as possible, given a certain limit on the effort to be expanded. The algorithms include the facility for varying this limit, producing anything from an algorithm requiring a single function evaluation to one doing an exact linear search. They are based on univariate minimization algorithms which we present first. These are normally at least quadratically convergent when derivatives are used and superlinearly convergent otherwise, regardless of whether or not the function is differentiable at the minimum.

Table of contents.

1. Introduction.

Descent methods for minimizing a function $F(x)$, $x \in E^n$, normally construct a sequence of estimates $\{x^{(k)}\}$ to the minimum such that

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)}$$

with

$$F(x^{(k+1)}) < F(x^{(k)}) \quad .$$

The vector $p^{(k)}$ is known as the direction of search and $\alpha^{(k)}$ as the steplength. Gill and Murray [1] describe a reliable and efficient algorithm for-determining the steplength $\alpha^{(k)}$ in the case that $F(x)$ is continuously differentiable. In this report we show how to construct a steplength algorithm that is equally efficient when minimizing certain classes of nondifferentiable functions.

We will at first restrict ourselves to the case that $F(x)$ is either

$$F_S(x) = \sum_{i=1}^{m} \max(0, f_i(x))$$

or (1.1)

$$F_M(x) = \max_{1 \le i \le m} f_i(x)$$

where the functions $\{f_i3$ are of the form

$$f_i : E^n \rightarrow E^1$$

and are continuously differentiable. We assume that every time the function $F(x)$ is evaluated the values $f_1(x),\ldots,f_m(x)$ are also made available. In Section 5 we will indicate how the algorithms developed for $F_S(x)$ and $F_M(x)$ could be modified to make them applicable to a wider class of functions. Basically, the ideas to be described here could be modified to construct a steplength algorithm for any nondifferentiable function $F(x)$ with the following properties:

(1) $F : \Gamma \subseteq E^n \to E^1$ is continuous on Γ.

(2) The directional derivative $F'(x)(\eta) =$

$$\lim_{h\to 0} \frac{1}{h} (F(x + h\eta) - F(x))$$

exists for all $\eta \neq 0$ everywhere on Γ .

(3) Any point z where the derivative $F'(z)$ does not exist satisfies an equation

$$\varphi^{(j)}(z) = 0$$

for some j , $1 \leq j \leq J$, where the functions $\varphi^{(1)},\ldots,\varphi^{(J)}$: $\Gamma \subseteq E^n \to E^1$ are known and are continuously differentiable.

In order to prove convergence for descent methods for multivariate minimization the steplength must satisfy certain criteria. Such criteria do not in general define a unique point, but a spectrum of values. Although all points in a particular range may satisfy the criteria equally well, they are not necessarily all of equal merit with regard to the efficiency of the method. Usually if two steps both meet the convergence criteria, the one that achieves the greatest reduction in $F(x)$ is to be preferred. The better the step, in this sense, the fewer iterations usually required to obtain a satisfactory approximation to the minimum. The question we face, therefore, in designing a steplength algorithm is not merely how to choose a step which satisfies the required criteria for convergence, but of the many that do how do we choose a "good" step. At the same time, however, how good a step we can choose depends on how much effort we are willing to expend, and hence all the algorithms we present contain the facility for varying this limit on the expense, producing anything from an algorithm requiring a single function evaluation to one doing an exact linear search.

2. Notation.

As explained in the last section, we shall initially be concerned with minimizing functions of the form

$$F_S(x) = \sum_{i=1}^{m} \max\,(0, f_i(x))$$

or

$$F_M(x) = \max_{1 \leq i \leq m} f_i(x)$$

where the functions $f_i(x)$ are continuously differentiable on E^n. We denote the gradient vectors of $f_i(x)$ by $v\ f_i(x)$ and define

$$g(x) = \sum_{i:\ f_i(x) > 0} \nabla f_i(x)\ , \quad \text{in the case that } F = F_s\ ,$$

or

$$g(x) = \nabla\ f_{j(x)}(x)\ , \text{ where } j(x) \text{ is the smallest index such that } F(x) = f_{j(x)}(x)\ , \text{ in the case that } F = F_M\ .$$

Note that $g(x)$ is the gradient of $F(x)$ wherever the latter is defined, and is one of the directional derivatives of $F(x)$ at the points where the gradient is not defined.

Let x be the current iterate and p be the direction of search along which the ste-p is to be taken. We are now omitting the superscript k for simplicity. Then it is convenient to write

$$f_i(\alpha) = f_i(x \vdash \alpha p) , \qquad i = 1,\ldots,m$$

and

$$F(\alpha) = F(x + \alpha p) .$$

It should always be clear form the context whether we are thinking of f_i or F as a function of a vector or of a steplength value α. Then

$$F(\alpha) = \sum_{i=1}^{m} \max (0, f_i(\alpha)) \qquad \text{in the case that } F = F_S$$

or (2.1)

$$F(\alpha) = \max_{1 \le i \le m} f_i(\alpha) \qquad \text{if} \quad F=F_M .$$

We denote the derivative of $f_i(\alpha)$, which is the projected gradient of $f_i(x + \alpha p)$ along p , by

$$f_i'(\alpha) = v f_i (x + \alpha p)^T P$$

and-denote the left and right derivatives of $F(\alpha)$ by $F'(\alpha)$ and $F_+'(\alpha)$ where

$$F'(\alpha) = \lim_{h \to 0-} \frac{F(\alpha+h) - F(\alpha)}{h}$$

and $F_+'(\alpha)$ is analagously defined.

We also define

$$F'(\alpha) = g(x + \alpha p)^T p .$$

Thus

$$F'(\alpha) = \sum_{i:\ f_i(\alpha) > 0} f_i'(\alpha) \quad , \quad \text{in the case that } F = F_S \tag{2.2}$$

or

$$F'(\alpha) = f'_{j(\alpha)}(\alpha)$$

where $j(\alpha)$ is the smallest index such that

$$F(\alpha) = f_{j(\alpha)}(\alpha) , \qquad \text{if } F = F_M .$$

Furthermore, $F'(\alpha)$ is the derivative of $F(\alpha)$ wherever the latter is defined and otherwise is either the left or the right derivative of $F(\alpha)$.

We also define $F''(\alpha)$ to be the second derivative of $F(\alpha)$ where it exists.

A point where the derivative of $F(\alpha)$ does not exist will be referred to as a point of derivative discontinuity, or just a discontinuity for short. Note that in the case $F = F_S$ the discontinuities occur at the zeros of the $f_i(\alpha)$. In the case $F = F_M$, the discontinuities z satisfy $F(z) = f_i(z) = f_k(z)$ for some $i \neq k$.

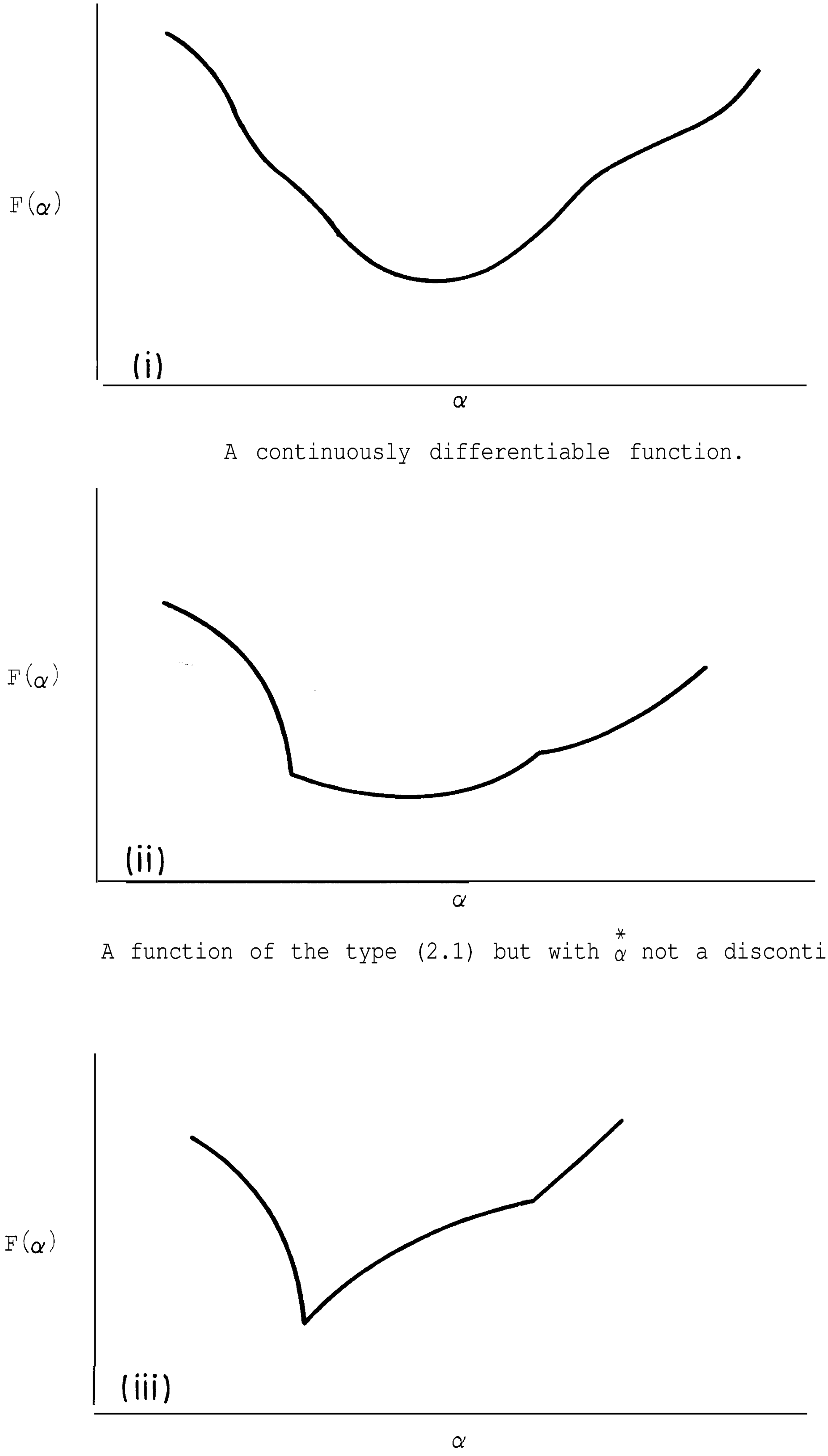
The term "convergence rate" will be used to mean the R-order of convergence in the sense of Ortega and Rheinboldt [2,p.290] . In each of our theorems, the corresponding stronger Q-order result also follows except in pathological cases, but as pointed out by Brent [3,p.35], it is often necessary to introduce rather artificial conditions to ensure this. Thus, for simplicity, we use only the R-order.

3. Univariate Minimization Methods.

In this section we present a number of algorithms to find a local minimum of the univariate function $F(\alpha)$, where $F(\alpha)$ is defined by (2.1). There is no loss of generality if we assume $F'(0) < 0$ and that there exists a local minimum $\overset{*}{\alpha} > 0$. This is a valid problem in its own right, but in developing our algorithms we shall bear in mind their use for constructing steplength algorithms for minimizing functions of several variables. This aspect of their use will be discussed in Section 4 .

The inappropriateness of using an algorithm which assumes $F(\alpha)$ is continuously differentiable can be seen by examining Figure 1 Efficient methods for functions with continuous derivatives usually determine $\overset{*}{\alpha}$ iteratively by successively approximating $F(\alpha)$ by a cubic or quadratic polynomial and taking as the next approximation to $\overset{*}{\alpha}$ the minimum of this approximating polynomial. The approximating polynomial is matched to $F(x)$ at the best known estimates to $\overset{*}{\alpha}$. It is quite clear in case (iii) that the minimum of an approximating polynomial may bear no relationship to $\overset{*}{\alpha}$. In case (ii) the approach is valid only if the approximating points all lie on the central portion of the curve. Such a situation is unlikely to be true initially when only a poor approximation to $\overset{*}{\alpha}$ is known. Consequently, the initial performance of such algorithms is poor even when the solution is not a discontinuity. It will be seen that from the point of view of steplength algorithms it is the initial performance which is crucial.

The two types of minima possible for functions of the type (2.1) illustrated in Figure 1 are emphasized in the statement of the necessary



A continuously differentiable function.

A function of the type (2.1) but with $\overset{*}{\alpha}$ not a discontinuity.

A function of the type (2.1) with $\overset{*}{\alpha}$ a discontinuity.

FIGURE 1

and sufficient conditions for $\overset{*}{\alpha}$ to be a minimum of $F(\alpha)$:

Necessary condition: Either $F'(\overset{*}{\alpha}) = 0$ with $F''(\overset{*}{\alpha})$ non-negative if it exists , or $\overset{*}{\alpha}$ is a derivative discontinuity of $F(\alpha)$ with $F'(\overset{*}{\alpha}) \leq 0$ and $F'_+(\overset{*}{\alpha}) \geq 0$.

Sufficient condition: Either $F'(\overset{*}{\alpha}) = 0$ and $F''(\overset{*}{\alpha})$ exists and is positive, or $\overset{*}{\alpha}$ is a derivative discontinuity of $F(\alpha)$ with $F'(\overset{*}{\alpha}) < 0$ and $F'_+(\overset{*}{\alpha}) > 0$.

Two algorithms are described (one with low overhead and one with higher overhead) with two variants (one which utilizes derivatives and one which does not). The essential feature of all the algorithms is that at each iteration the option of converging to either type of minimum is kept open. The step taken may be an estimate of either type, depending on which is considered more likely and/or prudent. When $\overset{*}{\alpha}$ is not a discontinuity the higher overhead algorithms are comparable in efficiency (in terms of the number of function evaluations) to the algorithms of [1] applied to just the continuously differentiable function which coincides with $F(\alpha)$ near $\overset{*}{\alpha}$. When $\overset{*}{\alpha}$ is a discontinuity the algorithms are comparable in efficiency to efficient rootfinding algorithms applied directly to the function of which $\overset{*}{\alpha}$ is a root.

Before describing the new algorithm in detail we shall review the algorithms of [1]. It is worth noting that because of the safeguards built into them they will work even if $F(\alpha)$ is not differentiable although for the reasons mentioned earlier their performance will usually be poor.

3.1 The Differentiable Case.

Here we present the basic ideas of the algorithms of Gill and Murray [1] for the univariate minimization problem when $F(\alpha)$ is differentiable. These ideas will be needed in Section 3.2. We do not make any attempt to present all the details of the algorithms and refer the reader to [1] for these.

We assume that at each iteration of the minimization we know an interval [a,b] in which the minimum $\overset{*}{\alpha}$ is known to lie. The interval [a,b] is called the interval of uncertainty and the points a and b are said to bracket the minimum. (Initially a is zero, but b is unknown. This situation is handled later).

Two cases-are treated, the case where both the function values $F(\alpha)$ and the derivatives $F'(\alpha)$ are used, and the case where the function values only are used. The latter case may occur either because the derivatives are not available, or because they are relatively expensive to compute. In the first case we assume that two points x and w are also known. The point x is the lowest point obtained so far, (i.e. $F(x) < F(\alpha)$ for any value of α at which the function has been evaluated), and w is either the second lowest or the last evaluation point. There are four possible configurations for x, w, a and b :

$$\begin{array}{lll} \text{(i)} & x = a & \text{and} \quad w<a \\ \text{(ii)} & x = a & \text{and} \quad w=b \\ \text{(iii)} & x = b & \text{and} \quad w = a \\ \text{(iv)} & x = b & \text{and} \quad w > b\ . \end{array} \tag{3.1}$$

In the case that the derivatives are not used, we assume that three points x, w and v are known, where x is the lowest point obtained so far, w is the second lowest, and v is either the third lowest or else is the most recent point. Then $a < x < b$ and the possible values for w and v are :

$$\begin{array}{llll} \text{(i)} & w = a & \text{and} & v < a \\ \text{(ii)} & w = a & \text{and} & v = b \\ \text{(iii)} & w = b & \text{and} & v = a \\ \text{(iv>} & w = b & \text{and} & v > b \end{array} \qquad (3.2)$$

The basic strategy of the algorithm is to use successive polynomial interpolation with safeguards. (We use interpolation to mean actually either extrapolation or interpolation). Thus at each iteration, a new point $\overset{A}{u}$ is chosen as the minimum of a polynomial approximating $F(\alpha)$ at some of the points already evaluated. Provided $\hat{u}$ satisfies certain conditions, u is set to $\hat{u}$, but otherwise the point is rejected and u is set to something different. The function is then evaluated at u . In the case that derivatives are used, the polynomial is a cubic chosen to agree in both function value and derivative with $F(\alpha)$ and $F'(\alpha)$ at the points x and w , and in the case that derivatives are not used, a quadratic is chosen to agree in function value with $F(\alpha)$ at the 'points x, w and v . Then u is set to $\hat{u}$ except in the following situations:

(i) $\hat{u}$ lies outside the interval of uncertainty [a,b]. This normally only occurs in cases (i) or (iv) of (3.1) or (3.2) when the step from x to $\hat{u}$ is an extrapolation step. However, because of round-off error, it could occur even in an interpolation step. Whenever $\hat{u}$ does not lie in [a,b] it is rejected and u is instead set to a point $\bar{u}$ obtained by a function comparison method which is guaranteed to lie in [a,b] . The function comparison method used is somewhat complicated and will not be described here.

(ii) $\hat{u}$ is obtained by extrapolation and although $\hat{u}$ lies in [a,b], $\bar{u}$ lies between $\hat{u}$ and x . In this case too the point is rejected and u is set to $\bar{u}$ instead. By extrapolation we mean that case (i) or (iv) applies in (3.1). Justifying this would require going into details about the function comparison method, but basically if $\hat{u}$ is not close to the best two points but close to a known poor point some change would seem warranted.

(iii) The step from x to $\hat{u}$ is greater in magnitude than half of the step taken at the iteration before last. Here too u is instead set to $\bar{u}$. The purpose of this restriction is to ensure that the algorithm does not produce a sequence of points oscillating back and forth at each iteration and reducing the interval of uncertainty by very little.

(iv) The point $\hat{u}$ lies too near one of the points already evaluated. In this case u is instead set to another point which is separated from those already evaluated by at least a certain tolerance tol(x) defined by

$$\text{tol}\ (\alpha) = \epsilon\,|\alpha| + \tau \qquad (3.3)$$

The function is then evaluated at u and the various points are updated as follows:

```
Case with derivatives:
     If $F(u) \leq F(x)$ then
          if $F'(u) \leq 0$ then $a \leftarrow u$ otherwise $b \leftarrow u$
          $w e x$ and $x \leftarrow u$
     otherwise
          if $u < x$ then $a \leftarrow u$ otherwise $b \leftarrow u$
          $w e u$ .

Case without derivatives:
     If $F(u) \leq F(x)$ then
          if $u \geq x$ then $a \leftarrow x$ otherwise $b \leftarrow x$
          $v \leftarrow w$ , $w \leftarrow x$ and $x \leftarrow u$
     otherwise
          if $u < x$ then $a \leftarrow u$ otherwise $b \leftarrow u$
          if $F(u) < F(w)$ then $v \leftarrow w$ and $w \leftarrow u$
          otherwise $v \leftarrow u$ .
```

This completes the description of one iteration of the algorithm.

3.1.1. The initial strategy.

The initial case is normally handled by specifying in advance a step length α_0 to try first. This then gives two initial values 0 and α_0 for x and w . In the case without derivatives, the second step must also be handled specially, using only the two points for the polynomial approximation, but we do not consider the details here. The point a is initially set to zero, but it may be several iterations before we determine an upper bound on the interval of uncertainty. The strategy in this case is to use polynomial extrapolation, just as in the case where b is known and situation (i) in (3.1) or (3.2) applies, but with u being set to $\hat{u}$ except in the following cases :

(i) the step from x to $\hat{u}$ goes across $\bar{\bar{u}}$, where $\bar{\bar{u}}$ is obtained by taking a step from x which is four times the step taken in the previous iteration. Here u is set to $\bar{\bar{u}}$.

(ii) The point $\hat{u}$ lies too close to or beyond a fixed upper bound on α beyond which we are not permitted to evaluate the function. Here u is set to a permissable point instead.

The function is then evaluated at the new point u and the other points are updated as in the case that b is known.

3.1.2. Convergence Criteria.

We complete the basic description of the algorithms by specifying that they terminate when the minimum is bracketed and

$$b - a < 2 \, \mathrm{tol}(x) \qquad \text{(case with derivatives)}$$

or

$$\max(x-a, b-x) < 2 \, \mathrm{tol}(x) \qquad \text{(case without derivatives)}$$

where tol(x) is defined by (3.3).

3.1.3. Convergence Results.

Here we state the convergence results for what we may call the theoretical procedures associated with the algorithms described above. By this we mean the algorithms with exact arithmetic applied to the exact function $F(\alpha)$, with the tolerances ϵ and τ set to zero. Since this might produce a zero step from x to $\hat{u}$, we also specify that if this happens u is instead set to $\bar{u}$ as defined by the function comparison method. We assume that an upper bound $\bar{b}$ has been found on the interval of uncertainty.

We also assume that $F(\alpha)$ is continuous on $[0,\bar{b}]$. (We can obtain convergence results even if $F(\alpha)$ is not differentiable, althought not the same rate of convergence). Let us define a stationary-inflection point as a point α, where $F'(\alpha)$ exists and equals zero, and which is neither a local maximum nor a local minimum of $F(\alpha)$. We also define a' generalized stationary-inflection point as a point α where either $F'_+(\alpha) = 0$ or $F'(\alpha) = 0$ and which is neither a local maximum nor a local minimum of $F(\alpha)$. Note that by a local minimum we mean either a weak or a strong local minimum, i.e. a point $\overset{*}{\alpha}$ such that $\exists\ \delta > 0$ s.t. $F(\alpha) \geq F(\overset{*}{\alpha})$ for $|\alpha - \overset{*}{\alpha}| < \delta$. We then have the following result:

Theorem 1.

Let $\{u_k\}$ be the sequence of points u generated by the theoretical procedures. Then in both the cases with and without derivatives the sequence $\{u_k\}$ converges to a point $\overset{*}{\alpha}$ which is either a local minimum or a generalized stationary-inflection point of $F(\alpha)$ on $[0,\bar{b}]$. Furthermore, suppose that $F''(\overset{*}{\alpha})$ is positive, $u_k \neq \overset{*}{\alpha}$ for all k, and $F'''(\alpha)$ is Lipschitz continuous on [a,b] as defined by [2,p.63].

Then the asymptotic convergence rate is quadratic, in the case with derivatives, and is superlinear with order 1.324..., in the case without derivatives.

We note that it is possible to modify the theoretical algorithm in order to ensure convergence to a local minimum. In practice this additional complication is not warranted since numerically one cannot distinguish between a stationary point and a minimum.

In order to prove the theorem, we need several lemmas. In the following, we use u_k , x_k , w_k , a_k , b_k , etc. to denote the various points at iteration k .

Lemma 1.

Suppose that the sequence of points $\{u_k\}$ contains a subsequence $\{u_{j_k}\}$ with the property that $u_{j_k} = u_{j_k}$ for all k , i.e. all the points in the subsequence are generated by the function comparison method. Then the sequence $\{u_k\}$ converges to a point $\overset{*}{u}$, and $\overset{*}{u}$ is a local minimum or generalized stationary inflection point (LM or GSIP).

Proof.

The proof that the sequence converges follows from a property of the function comparison method that at each such step the length of the interval of uncertainty is multiplied by θ where $0 < \theta < 1$. The rules for updating the end points of the interval of uncertainty ensure that it always contains a LM or GSIP, so we have $a_k, b_k, u_k \rightarrow \overset{*}{u}$,and $\overset{*}{u}$ is a LM or GSIP.

Lemma 2.

Suppose 3 K such that u_k is set to the point $\hat{u}_k$ for all $k > K$, i.e. ultimately no function comparison steps are needed. Then the sequence $\{u_k\}$ converges to a limit $\overset{*}{u}$. Also $x_k \rightarrow \overset{*}{u}$ and $w_k \rightarrow \overset{*}{u}$.

Proof.

We must have that for all $k \geq K$:

$$|u_k - x_k| \leq \frac{1}{2} |u_{k-2} - x_{k-2}|$$

as otherwise the point $\hat{u}_n$ would be rejected. Assume without loss of generality that K is even. Then

$$|u_{2k} - x_{2k}| \leq (\frac{1}{2})^{k - \frac{K}{2}} | u_K - x_K|$$

and

$$|u_{2k+1} - x_{2k+1}| \leq (\frac{1}{2})^{k-\frac{K}{2}} | u_{K+1} - x_{K+1}| \quad .$$

For simplicity we write

$$|u_k - x_k| \leq (\frac{1}{2})^{\frac{k}{2}} C$$

where $C = 2^{-\frac{K}{2}} \max \{|u_K - x_K| , |u_{K+1} - x_{K+1}| \}$.

Since x_k is the lowest point so far and u_k is the new point, we always have either $x_k = u_{k\ 1}$, or $x_k = x_{k-1}$. Thus either there exists J such that $x_k = xJ$ for $k > J$, i.e. no subsequent point u_k is lower than x_J , or there is a subsequence $(x_{j_k}\}$ such that $x_{j_k} = u_{j_k-1}$ for all k . In the former case $u_k \to x_J$, so assume the latter. Consider u_{k+1}, u_k . Let j_ℓ be the largest element of the subsequence such that $j_\ell < k$, i.e. $x_k = x_{k-1} = \cdots = x_j = u_{j_\ell - 1_\ell}$. If $j_{\ell+1} = k + 1$, i.e. $x_{k+1} = u_k$, then

$$|u_{k+1} - u_k| = |u_{k+1} - x_{k+1}| \leq (\frac{1}{2})^{\frac{k+1}{2}} C \quad .$$

Otherwise $x_{k+1} = x_{j_\ell}$ and we have

$$|u_{k+1} - u_k| \le |u_{k+1} - x_{j_\ell}| + |u_k - x_{j_\ell}|$$

$$\le |u_{k+1} - x_{k+1}| + |u_k - x_k|$$

$$\le 2 . (\frac{1}{2})^{\frac{k}{2}} c$$

$$\le (\frac{1}{2})^{\frac{k}{2} - 1} c \quad .$$

Using this result, we have that for any $i > k$

$$|u_i - u_k| \le |u_i - u_{i-1}| + |u_{i-1} - u_{i-2}| + \ldots + |u_{k+1} - u_k|$$

$$\le (\frac{1}{2})^{\frac{k}{2} - 1} c \; (1 + \frac{1}{2} + \frac{1}{4} + . \, . . >$$

$$\le (\frac{1}{2})^{\frac{k}{2} - 2} C \quad .$$

Therefore for all δ there exists M such that for i,k >_M, $|u_i - u_k| < \delta$, and hence $\{u_k\}$ is a Cauchy sequence and converges. Clearly $\{x_k\}$ and $\{w_k\}$ also converge to the same point.

The following lemma is presented without proof.

Lemma 3.

If $\overset{*}{u}$ is not a LM or GSIP of $F(\alpha)$, then there exists $\delta > 0$ such that the interval $(\overset{*}{u} - 6, \overset{*}{u} + \delta)$ contains no LM or GSIP and such that cubic interpolation with at least one of the two points in the interval, or quadratic interpolation with at least two of the three points in the interval, is good enough that the minimum of the approximating polynomial lies outside the interval.

The following lemma is presented in a more general form than needed here so that it can be used in the next section.

Lemma 4.

Assume that there is a subsequence $(u_{j_k}\}$ such that each point in the subsequence is generated by polynomial approximation to $F(\alpha)$ with, in the cubic case, x_{j_k} as one of the two points where the fit is made, or, in the quadratic case, x_{jk} and w_{jk} as two of the three points. Then, if $\{u_k\}$, $\{x_k\}$ and $\{w_k\}$ all converge to $\overset{*}{u}$, $\overset{*}{u}$ is a LM or GSIP of $F(\alpha)$.

Proof.

Let 6 be that of Lemma 3. Since the sequences all converge to $\overset{*}{u}$, there exists K- such that u_k, x_k and w_k are all separated from $\overset{*}{u}$ by at most δ , for all $k \geq K$. Therefore the points used for the fit ultimately lie within δ of $\overset{*}{u}$. Thus the result follows, since otherwise we can apply Lemma 3 to show that the new point u_k satisfies $|u_k - \overset{*}{u}| > \delta$, which is a contradiction.

Proof of Theorem 1.

Either there is a subsequence $\{u_{j_k}\}$ all generated from function comparison steps, or there exists K such that $u_k = \hat{u}_k$ for $k \geq K$. In the former case the first part of the result follows from Lemma 1. In the second case we can apply Lemmas 2 and 4 to conclude that $u_k \to \overset{*}{u}$, and. $\overset{*}{u}$ is a LM or GSIP. This completes the proof of the first part of the theorem, and we write $\overset{*}{\alpha} = \overset{*}{u}$.

If the hypotheses of the second part of the theorem hold, we can conclude that ultimately the points will be generated by successive polynomial approximation alone. The superlinear convergence of order 1.324..., for successive quadratic interpolation was shown by Jarratt [4] in 1967, and by Kowalik and Osborne [5, p.20] in 1968. In 1973, Brent [3,p.35] showed that their results for the Q-order (see Section 2) were not true in certain pathological cases and showed how to introduce extra assumptions to avoid these. He also showed that the R-order is at least 1.324... in all cases. The rate of convergence for successive cubic interpolation was shown to be quadratic (again except in pathological cases) by Overton [6], using the symbolic manipulation system MACSYMA [7]. This was independently rediscovered (also using MACSYMA) and a considerably simplified proof for the R-order was presented by Bjorstad and Nocedal [8].

Clearly the safeguards (i) to (iii) will no longer be involved once the quadratic or superlinear convergence sets in. This completes the proof of the theorem.

Note that it does not make sense to talk about just any local minimum of the function $\hat{F}(\alpha)$ which approximates $F(\alpha)$ by calculating it on a finite machine, since such a function is really just a step.-function and may have a lot of local minima very close together (see Brent [3,p.63]). Instead, we can say that the algorithm produces an approximate local minimum in the following sense:

Theorem 2.

The algorithm applied to the computed function $\hat{F}(\alpha)$ using the (computed) derivatives $\hat{F}'(\alpha)$, terminates with points a and b such that a < b , and

(i) $\hat{F}'(a) \leq 0$ and $\hat{F}'(b) > 0$ or $\hat{F}'(a) \leq 0$ and $\hat{F}(b) > \hat{F}(a)$

or $\hat{F}''(b) > 0$ and $\hat{F}(a) > \hat{F}(b)$

and

(ii) $\text{tol}(x) \leq b - a \leq 2\ \text{tol}(x)$

(where $x = a$ if $\hat{F}(a) < \hat{F}(b)$ and otherwise $x = b$) ,

Theorem 3.

The algorithm applied to the computed function $\hat{F}(\alpha)$ without derivatives terminates with points a, x and b such that $a < x < b$ and

(i) $\hat{F}(a) \geq \hat{F}(x)$ and $\hat{F}(x) \leq \hat{F}(b)$

(ii) $\text{tol}(x) \leq \max(x-a, b-x) \leq 2\ \text{tol}(x)$.

As long as the tolerances ϵ and τ are chosen so that tol (x) is a reasonable minimum distance to require between two points before comparing their function values, then the above is as near as we can get to giving conditions for a "reasonable" local minimum to satisfy. The results are easily verified by examining the algorithm.

3.2. The Nor-differentiable Case.

In this section we describe the changes that must be made to the algorithm described in Section 3.1 to create our new algorithm for univariate minimization when $F(\alpha)$ is given by (2.1). It is necessary to modify only one part of the algorithm, namely the method used for selecting the point $\hat{u}$. The safeguards which when necessary reject $\hat{u}$ and set u to another point, and the method for updating the points a,b,x,w and v, are all left unchanged. As mentioned earlier, the key strategy of the new algorithm is to try to recognize whether $\overset{*}{\alpha}$ is a discontinuity or not, and to then select $\hat{u}$ accordingly as either a direct estimate of the discontinuity, or as an estimate of the minimum of a polynomial approximating $F(\alpha)$.

Several different cases are treated. We have already noted that we are concerned with the two possibilities $F(\alpha) = F_S(\alpha)$ and $F(\alpha) = F_M(\alpha)$ as defined by (2.1) . These will both be described together as far as possible. We also consider both the case where the function values $f_i(\alpha)$ and the derivatives $f_i'(\alpha)$ are used and the case where function values only are used. Recall that we have extended the definition of a derivative by defining $F'(\alpha)$ in (2.2) . For simplicity we initially confine our attention to the case with derivatives. Finally we describe two versions of the algorithm, a low overhead version and a higher overhead version. The latter makes much more use of all the information known but requires more operations to choose the new point $\hat{u}$. The two versions have similar asymptotic convergence properties but the higher overhead version should be more efficient in terms of the number of function evaluations required to obtain some specified accuracy (especially for low accuracy requirements). The difference between the two methods is likely to be more significant the higher the number of discontinuities is. In most applications the computer time is dominated by the time spent evaluating the function, so the higher overhead version is expected to be much the more useful in practice. However, for simplicity we describe the low overhead version first.

3.2.1 The Low Overhead Version.

As in Section 3.1, we assume that at each iteration we have an interval of uncertainty [a,b] and points x and w satisfying (3.1). The process for determining $\hat{u}$ may be divided into a number of parts:

(i) The minimum and maximum discontinuities contained in the interval $[a,b]$ are estimated. In the case that $F = F_S$ this is done as follows. The function values $f_i(a)$ and $f_i(b)$ are compared, for each i from 1 to m . If for some i , $f_i(a)$ and $f_i(b)$ have opposite signs then there is a discontinuity between a and b given by the zero of $f_i(\alpha)$. This is estimated by the Newton step from x . If z_i lies outside $[a,b]$ it is replaced by the secant estimate $(a\, f_i(b) - b\, f_i(a))\, / \,(f_i(b) - f_i(a))$ which is guaranteed to lie in $[a,b]$. After this has been done for all i , z_L is set to the minimum of the z_i and z_R is set to the maximum (it is not necessary to store the z_i ; z_L and z_R can be updated as each z_i is computed). If there were no discontinuities located between a and b , i.e. $f_i(a)$ and $f_i(b)$ had the same sign in every case, z_L is set to b and z_R is set to a . Note that comparing $f_i(a)$ and $f_i(b)$ for all i will identify all the discontinuities between a and b if the functions f_i are sufficiently near linear, although it may not identify them all in general, since a function may have a zero in $[a,b]$ and still have the same sign at a and b .

In the case that $F = F_M$ the discontinuities z are no longer given by $f_i(z) = 0$, but by $f_i(z) = f_k(z) = F(z)$ for some $i \neq k$. The estimate z_L of the minimum discontinuity is then made as follows. Let $j(a)$ be as defined in (2.2), i.e. normally $j(a)$ is the index of the only function which has the largest value at a . Then for each $i \neq j(a)$ the zero of $f_{j(a)}(\alpha) - f_i(\alpha)$ is estimated by the Newton step from a , i.e. $y_i^{(a)}$ is set to $a - [f_{j(a)}(a) - f_i(a)] \,/\, [f'_{j(a)}(a) - f'_i(a)]$. Note that there is no reason to suppose that $y_i^{(a)}$ lies in $[a,b]$. Then k_L and z_L are defined by

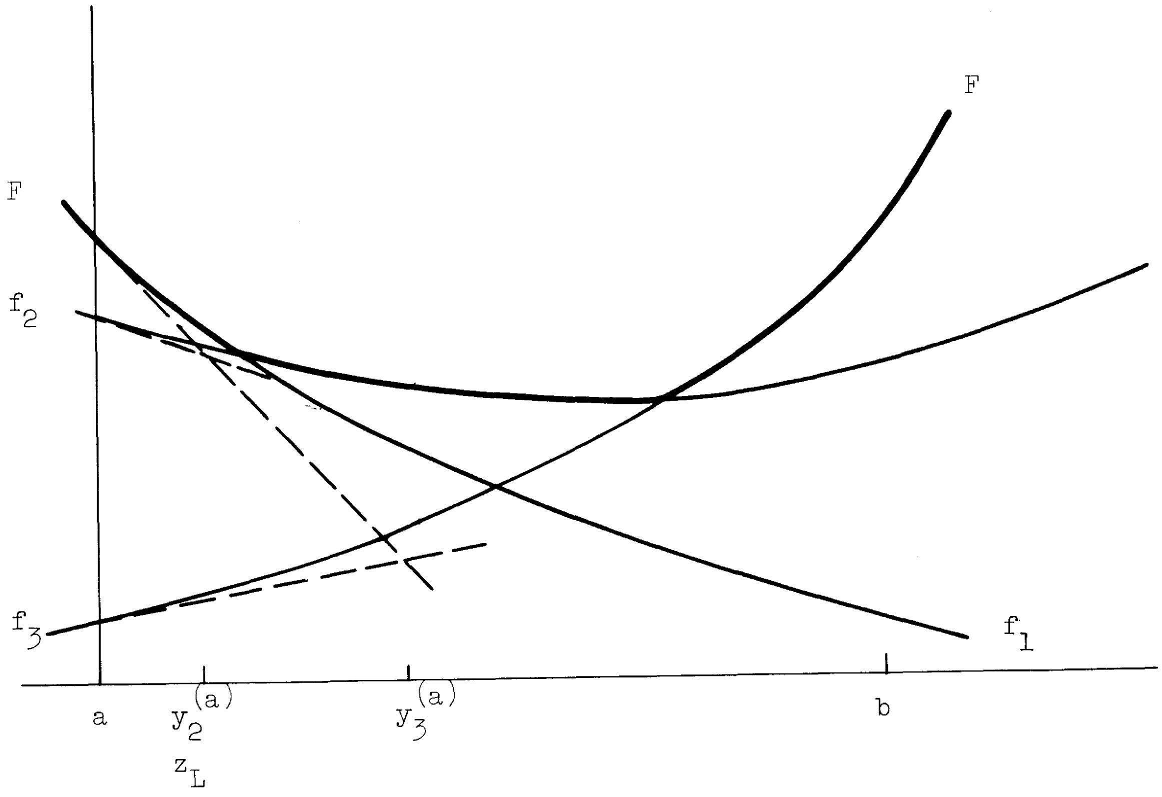
$$z_L = \min \{y_i^{(a)} : y_i^{(a)} > a\} = y_{k_L}^{(a)}$$

If none of the $y_i^{(a)}$ are greater than a then z_L is set to b and k_L is undefined. The estimate z_R of the maximum discontinuity and the index k_R are similarly defined by looking at the Newton step from b to the zero of $f_{j(b)}(\alpha) - f_i(\alpha)$ for each i . Note that it is not necessary to store all the $y_i^{(a)}$ or $y_i^{(b)}$. Figure 2 illustrates the process.

<u>However,</u> if F = FM , and $k_L = j(b)$ and $k_R = j(a)$, indicating that there is only one discontinuity in [a,b], then <u>in all subsequent computations of $\hat{u}$</u> part (i) is omitted and z_L and z_R are set to $\bar{z}$ as defined below. The reason for this is simply to avoid estimating the zeros of all the other $f_{j(a)}(\alpha) - f_i(\alpha)$ and $f_{j(b)}(\alpha) - f_i(\alpha)$ when it is unlikely that any of them will have any relevance. Note that this is the only place in the algorithm where any information need be retained from previous iterations other than a,b,x,w and the function and derivative values.

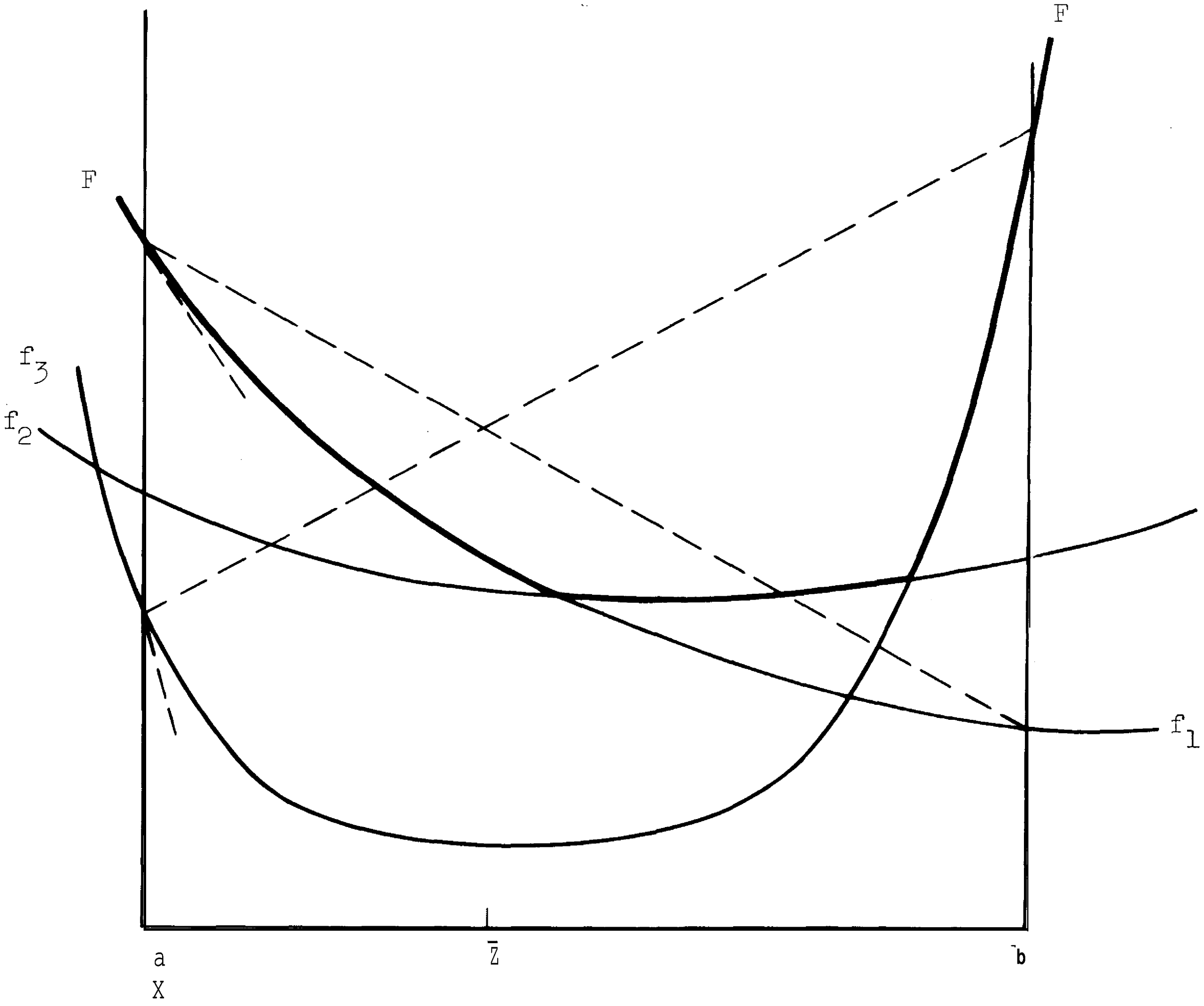
(ii) A point $\bar{z}$ is defined as follows. In the case $F = F_S$, $\bar{z}$ is defined to be the average of all the estimates z_i of the discontinuities located in [a,b]. In the case F = FM, $\bar{z}$ is set to an estimate of the zero of $f_{j(a)} - f_{j(b)}$ which must lie in [a,b]. The same technique used in (i) for estimating the zero of $f_i(\alpha)$ is used, i.e. first the Newton estimate from x is tried, and if this lies outside [a,b] it is replaced by the secant estimate using a and b . This is illustrated in Figure 3.

If there do not appear to be any discontinuities in [a,b], i.e. $F = F_S$ and none of the $f_i(\alpha)$ differ in sign at a and b , or $F=F_M$ and j(a) = j(b) , then $\bar{z}$ is undefined. If $F = F_M$ and $\bar{z}$ is defined then we insist that $z_L \leq z \leq z_R$ by setting $z_L = \min(z_L, \bar{z})$ and $z_R = \max(z_R, \bar{z})$. This may be necessary because of the different methods for making the three estimates.



Estimating the leftmost discontinuity in [a,b] for $F = F_M$.

FIGURE 2



. Determining $\bar{z}$ for $F = F_M$. The zero of $f_1 - f_3$ is estimated. First the Newton step from x is tried, but since this lies outside [a,b] it is replaced by the secant step using a and b .

FIGURE 3

(iii) If w is not equal to a or b , i.e. situation (i) or (iv) in (3.1) applies, then we wish to estimate whether there are any discontinuities between w and x . (If w equals a or b this has already been done in (i)). In the case that $F = F_S$, this is done by comparing $f_i(w)$ with $f_i(x)$ for each i and seeing whether they have opposite sign for any i . This can be done at the same time as the z_i are estimated in (i) . In the case that $F = F_M$ it is done simply by seeing whether j(w) and j(x) are equal. No attempt is made to estimate any discontinuities.

(iv) Let us introduce some new notation. For a given point y we define

$$F^{(y)}(\alpha) = \sum_{i:f_i(y) > 0} f_i(\alpha) \qquad \text{if } F = F_S$$

$$\text{or} \qquad F^{(y)}(\alpha) = f_{j(y)}(\alpha) \quad , \qquad (3.4)$$

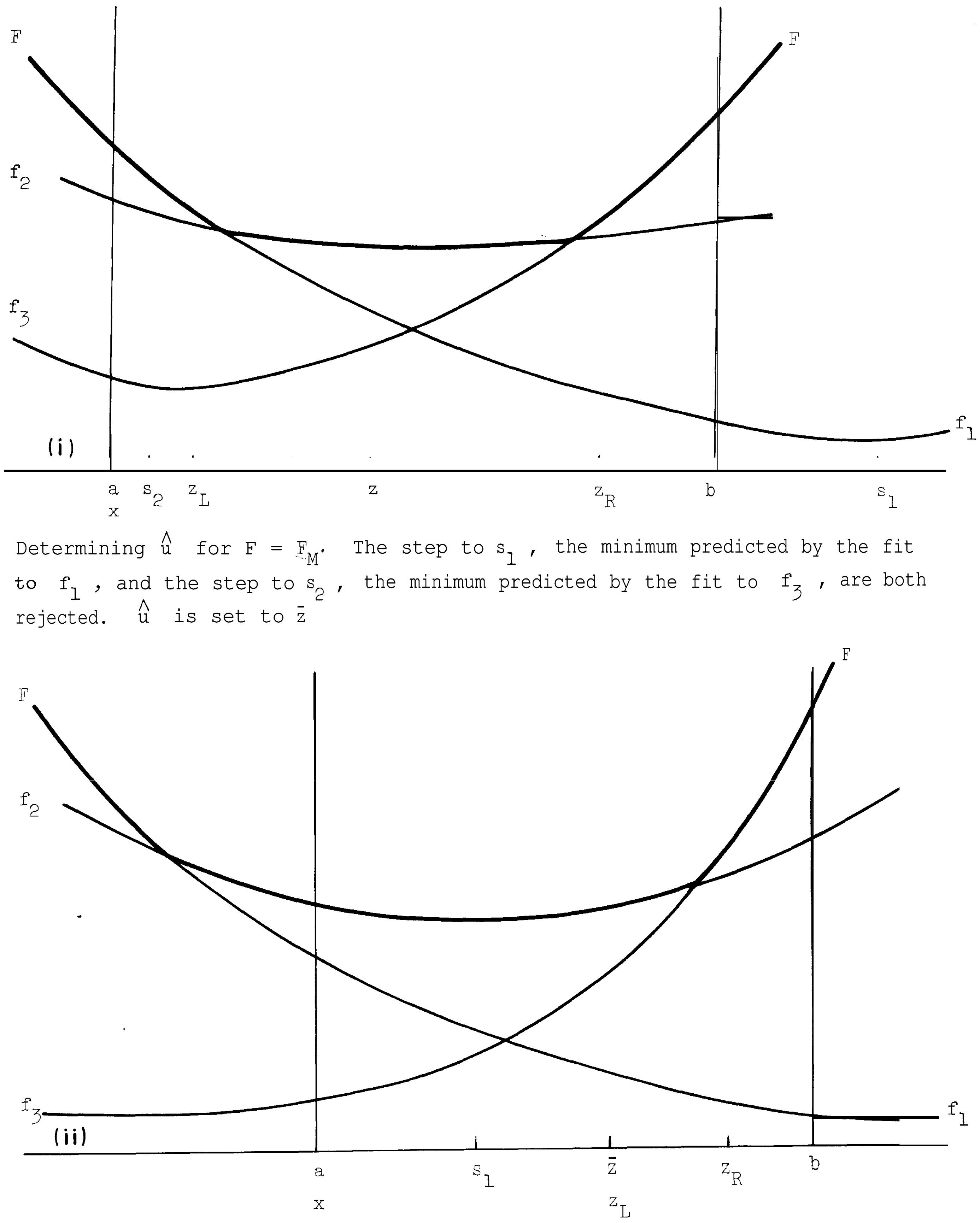
where $j(\alpha)$ is defined by (2.2) if $F=F_M$.

Then $F^{(y)}(\alpha)$ is a continuously differentiable function coinciding with $F(\alpha)$ in the interval containing y over which $F(\alpha)$ is differentiable. We denote the derivative of $F^{(y)}(\alpha)$ by $F^{(y)'}$ (cd .

In this part then we compute the values $F^{(a)}(b)$, $F^{(a)'}(b)$, $F^{(b)}(a)$ and $F^{(b)'}(a)$. Again for $F = F_S$ this can be done at the same time as the computation of the z_i in (i) .

(v) We are now ready to make our first polynomial approximation. The idea here is to fit a polynomial to a differentiable function coinciding with $F(\alpha)$ in a certain interval and to take the minimum of the polynomial as the new point u only if the step to it does not cross any of the estimates of the discontinuities. The ultimate quadratic convergence rate for successive cubic approximation quoted in Section 3.1.1 holds only if the approximation is made at the lowest points available, which do not necessarily bracket the minimum, Thus ultimately we would like each point $\overset{A}{u}$ to be obtained by approximating $F(\alpha)$ at x and w in the case that $\overset{*}{\alpha}$ is not a discontinuity. Therefore if there were no discontinuities located between w and x in (iv), or in (i) if $w=a$ or $w = b$, the point s_1 is computed as the minimum of the cubic fitted to $F(\alpha)$ at x and w, i.e. agreeing in function value and derivative with $F(\alpha)$ and $F'(\alpha)$ at x and w. However, if there was at least one discontinuity located between x and w, then s_1 is computed as the minimum of the cubic fitted to $F^{(x)}(\alpha)$ at a and b using the values in (iv) (recall $x=a$ or $x=b$). The reason that a and b are used rather than x and w is that this choice of fit cannot impede the ultimate rate of convergence in the case that $\overset{*}{\alpha}$ is not a discontinuity since then eventually there can be no discontinuities located between x and w. It is our view that this strategy,using an interpolation fit instead of an extrapolation fit while still not near the solution, is slightly more reliable than if the fit was made to $F^{(x)}(\alpha)$ at x and w regardless of whether there were discontinuities between the two points.

If s_1 lies in $[a,b]$ and the step from x to s_1 does not cross any discontinuities, i.e. $a \leq s_1 \leq z_L$ if $x = a$ or $z_R \leq s_1 \leq b$ if $x = b$, then $\hat{u}$ is set to s_1. Otherwise the step is rejected. This is illustrated in Figure 4.



Determining $\hat{u}$ for $F = F_M$. The step to s_1 , the minimum predicted by the fit to f_1 , and the step to s_2 , the minimum predicted by the fit to f_3 , are both rejected. $\hat{u}$ is set to $\bar{z}$

The step to s_1, the minimum predicted by the fit to f_2, is accepted. $\hat{u}$ is set to s_1.

FIGURE 4

(vi) If the step is rejected in (v), a second cubic fit is tried, this time fitting the cubic to $F^{(b)}(\alpha)$ at a and b if x=a, or to $F^{(a)}(\alpha)$ at a and b if x=b . Thus for example in Figure 4 (i) , after the step to s_1 is rejected, the point s_2 is set to the minimum of the cubic interpolating $f_2(\alpha)$ at a and b . This second cubic fit may be of crucial importance to the algorithm's performanceaswill be described later. Then as in (v), if s_2 lies in [a,b] and the step from x to s_2 does not cross any discontinuities, $\hat{u}$ is set to s_2 . Otherwise this step is rejected too.

(vii) If the steps in (v) and (vi) have both been rejected, this implies that the step from a to the estimate of the minimum of the differentiable function coinciding with $F(\alpha)$ at a crosses the estimate of a discontinuity. The same is true of the step from b estimating the minimum of the differentiable function coinciding with $F(\alpha)$ at b . Hence the conclusion is drawn that $\overset{*}{\alpha}$ may be a discontinuity. Therefore $\hat{u}$ is set to $\bar{z}$ as defined in (ii).

This completes the description of the choice of u when an interval of uncertainty is known. We now describe the changes that must be made to the above when minimum has not yet been bracketed. We have x=a and w < x. Then $\hat{u}$ is defined as follows.

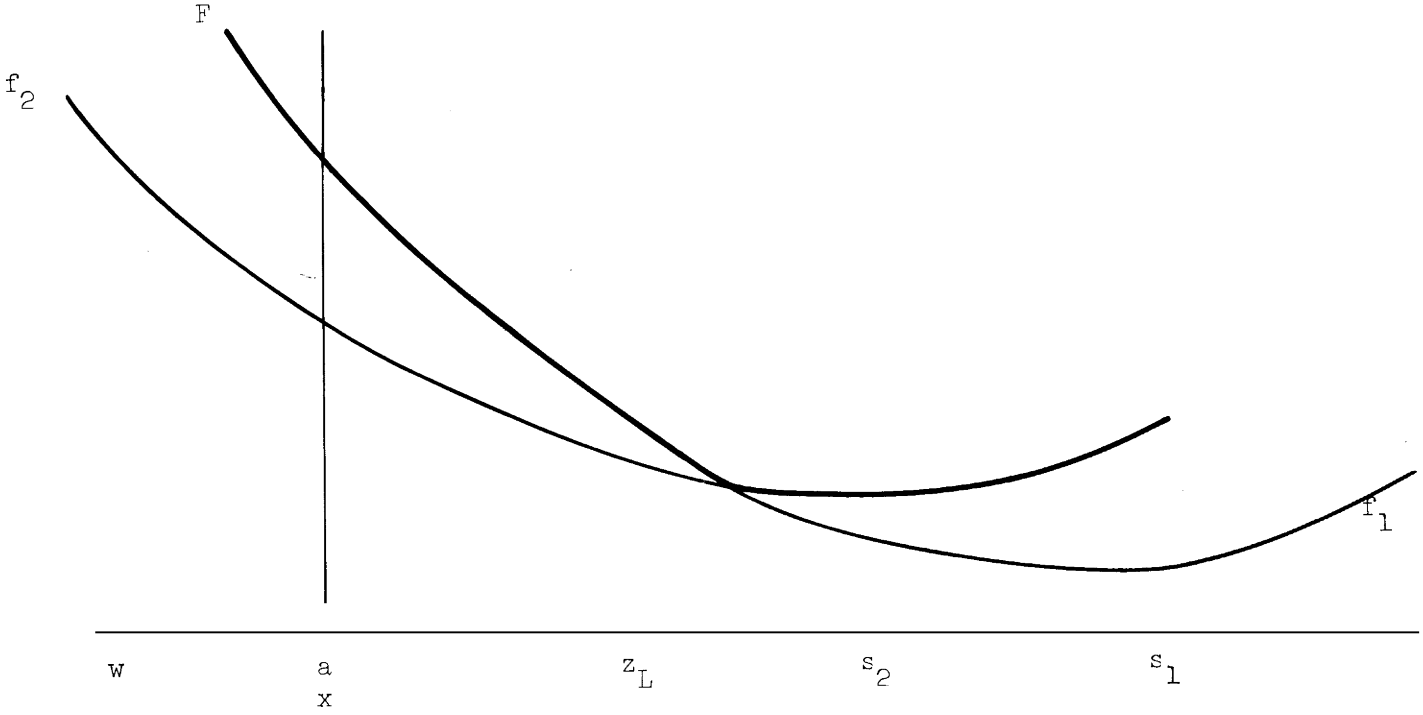
(i) Here the minimum discontinuity is estimated. If $F = F_S$ the zero of each of the $f_i(\alpha)$ is estimated by the Newton step from a , and z_L is set to the smallest estimate greater than a . If $F = F_M$, then z_L is defined as in the case where b is known.

(ii) and (iii) are omitted.

(iv) Here we compute $F^{(x)}(w)$ and $F^{(x)'}(w)$ as defined by (3.4).

(v) The point s_1 is computed as the minimum of the cubic fitted to $F^{(x)}$ at x and w . If $x \leq s_1 \leq z_L$, then $\hat{u}$ is set to s_1 ; otherwise the step is rejected.

(vi) If the step in (v) was rejected, a second cubic fit is made, this time fitting to the differentiable function which is thought to coincide with $F(\alpha)$ beyond the discontinuity which is estimated by z_L . This is done by noting in (i) which function it is whose zero is estimated by z_L . For example, if $F = F_S$, and z_L estimates the zero of $f_1(\alpha)$, and $f_1(x) > 0$, then the differentiable function thought to coincide with $F(\alpha)$ beyond this zero would be $F^{(a)}(\alpha) - f_1(\alpha)$. Consequently the value of this function and its derivative would be computed at x and w in order to make the cubic fit. An example in the case $F = F_M$ would be that z_L estimates the zero of $f_1(\alpha) - f_2(\alpha)$ and $F(x) = f_1(x)$. Then the differentiable function in question would be f_2 and the value of f_2 and its derivative at x and w would be used for the cubic fit. Let s_2 be the minimum of the cubic thus defined. If $s_2 \geq z_L$ we set $\overset{A}{u}$ to s_2 ; otherwise the step is rejected. The situation when F = FM is illustrated in Figure 5 .



Determining $\hat{u}$ for $F = F_M$ when the minimum is not bracketed. The step to s_1, the minimum predicted by the fit to f_1, is rejected, and so $\hat{u}$ is set to s_2, the minimum predicted by the fit to f_2.

FIGURE 5

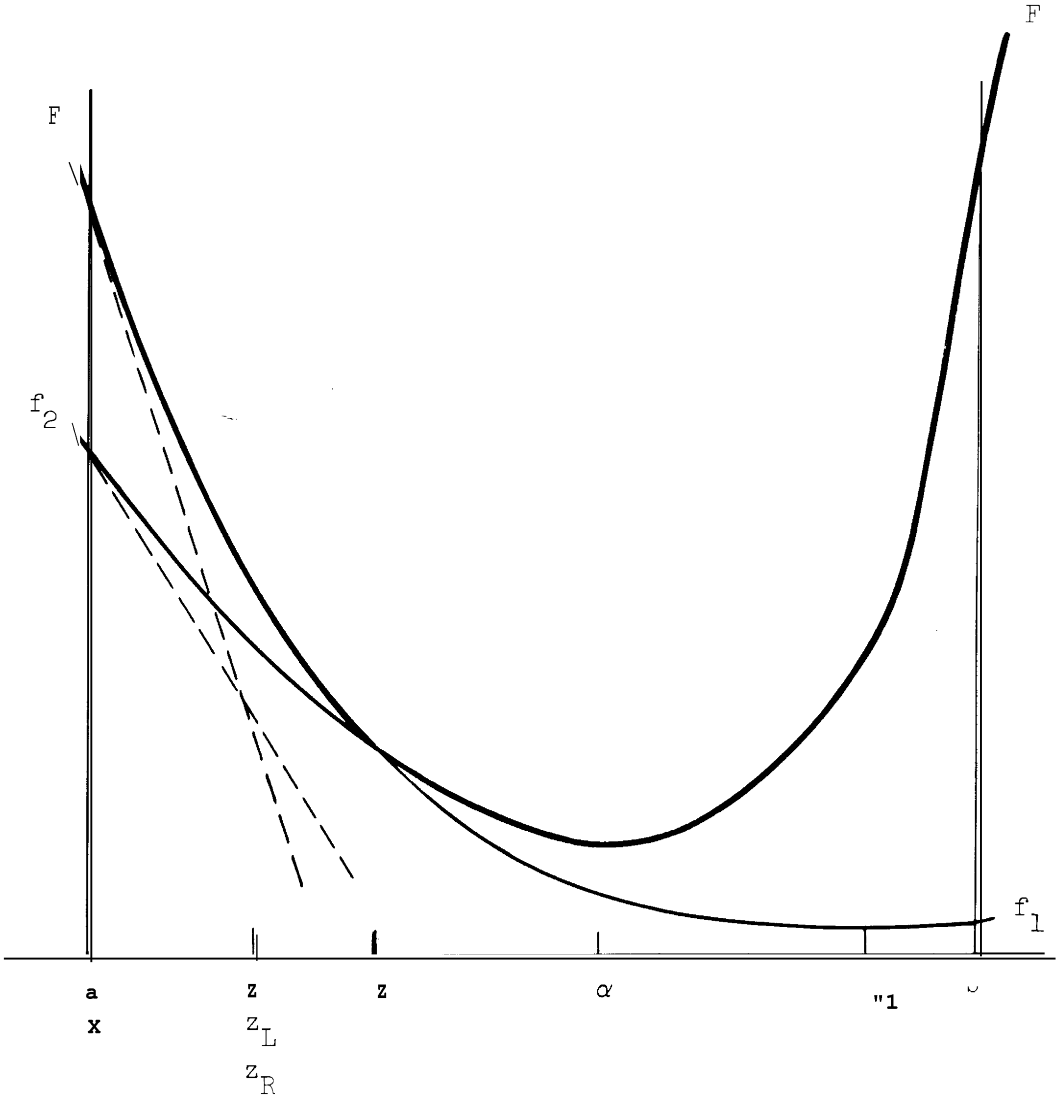
(vii) If the steps in (v) and (vi) were both rejected, this implies that the steps to the estimates of the minima of each of the differentiable functions coinciding with $F(\alpha)$ on either side of the discontinuity estimated by z_L both go across the dicontinuity. Hence we conclude that $\overset{*}{\alpha}$ is likely to be at the discontinuity and set $\hat{u}$ to z_L.

3.2.2 Comments on the Algorithm.

In the algorithm described the asymptotic rate of convergence will usually be quadratic, irrespective of whether or not $\overset{*}{\alpha}$ is a point of discontinuity. This is because ultimately the points generated will either be those resulting from successive cubic interpolation estimating the minimum of a differentiable function or from Newton's method estimating the zero of a different differentiable function, and both processes normally have a quadratic rate of convergence. Note that since $\bar{z}$ is an estimate of the average of the discontinuities we might expect the number of discontinuities between a and b to be halved at each step. Consequently even on problems for which there is a large number of discontinuities in the region of interest the number within the interval of uncertainly will soon become small.

To our knowledge the only other univariate minimization or line search algorithm which has been proposed for special nondifferentiable functions is that of Charalambous and Conn [9] for $F = F_M$. Their algorithm does not include the safeguards that we have described. Also, a basic iteration of their algorithm is quite different from ours in a number of ways. Suppose a is the lowest point x. Their algorithm estimates the zeros of $f_{j(a)} - f_i$ for each i by $y_i^{(a)}$ as ours does. It then similarly estimates the values of f_k at $y_i^{(a)}$ for each i and k and hence estimates the value of F at each $y_i^{(a)}$. The point $y_i^{(a)}$ with the lowest estimated value of F is then chosen for the new point and a function evaluation is made. Then a cubic interpolation step is taken only if the new function value is higher than $F(x)$. This approach is quite different from ours where $\hat{u}$ is set to a discontinuity estimate only if the step to the cubic interpolation estimate crosses a discontinuity estimate. Also we make cubic fits only to differentiable functions, i.e. to $F^{(a)}(\alpha)$ or $F^{(b)}(\alpha)$ rather than $F(\alpha)$. Since the points estimated by $y_i^{(a)}$ may not even be discontinuities, our higher overhead version (to be described in the next section) presents a better way to estimate the minimum supposing that is is at a discontinuity.

In some situations the algorithm of [9] may converge to a point of discontinuity which is not a minimum. This would also happen in our algorithm if step (vi) were omitted, i.e. if $\hat{u}$ were set to $\bar{z}$ without making a second cubic fit when the step to the estimate of the minimum of the first interpolating polynomial used in (v) crosses a discontinuity estimate. This is illustrated in Figure 6 for $F = F_M$. Here $z_L = z_R = \bar{z}$ as there is only one discontinuity between a and b and $\bar{z} < \overset{*}{z} < s_1$, where $\overset{*}{z}$ is the exact zero of $f_1 - f_2$. If no second fit is made in step (vi) but $\hat{u}$ is set to $\bar{z}$ the points generated will converge to $\overset{*}{z}$ if the points $\bar{z}$ converge to $\overset{*}{z}$ from the left. This will happen in this example if $(f_1''(\overset{*}{z}) - f_2''(\overset{*}{z})) \cdot (f_1'(\overset{*}{z}) - f_2'(\overset{*}{z})) < 0$ since Newton's method to find the zero $\overset{*}{z}$ of $\varphi(z)$ converges from the left if $\varphi''(\overset{*}{z})\, \varphi'(\overset{*}{z}) < 0$. Clearly what is needed is to generate a point between $\overset{*}{z}$ and b , and this is done by stepping to the minimum of the cubic fitted to f_2 at a and b in step (vi).



An example for $F = F_M$ where successively setting $\hat{u}$ to $\bar{z}$ because s_1 is rejected causes convergence to $\overset{*}{z}$ instead of $\overset{*}{\alpha}$.

FIGURE 6

An alternative strategy for avoiding this difficulty is to estimate whether the gradient of $F(\alpha)$ changes sign at the discontinuity estimated by $\bar{z}$, and to set $\hat{u}$ to $\bar{z}$ only if this happens. However, the second cubic fit is recommended since it can give a good estimate of the minimum at the same time as rejecting the estimate of the discontinuity. In any event if the gradient is thought not to change sign at the discontinuity some alternative step must be computed.

Another point worth noting is that it might seem that an almost as efficient algorithm could be designed saving some storage by not requiring the $f_i(w)$ to be available as well as $f_i(a)$ and $f_i(b)$ (and perhaps the corresponding derivatives). In fact saving $f_i(w)$ requires no extra storage as a third vector in addition to those for the function values at a and b is required anyway for the evaluation of the function values at the new point u , and since the new w is always either the old a or the old b , the function values at w can be retained by interchanging the new vector with the old vector that would otherwise be overwritten. Of course this is really only of academic interest since we do not expect storage of a vector of length m to be significant.

3.2.3 The Higher Overhead Version.

We now describe a second version of the algorithm which requires more housekeeping operations and/or storage , but makes fuller use of the information available. The basic difference between the two versions is that in the higher overhead version we do not restrict the number of cubic fits to one or two, but allow up to m cubic fits. Consequently $\hat{u}$ is always set to either the estimate of a minimum between two adjacent discontinuity estimates or to a specific discontinuity estimate. The other difference between the two versions is that we now estimate the discontinuities by inverse cubic interpolation at two points. Thus the estimate of the discontinuity is chosen as the zero of the inverse cubic which agrees with the inverse of $\varphi(\alpha)$ in both function and derivative values, where $\varphi(\alpha)$ is the function whose zero is desired. Inverse interpolation is preferable to direct interpolation for this purpose because the zero of the inverse cubic must be unique whereas the direct interpolating cubic may have several zeros. For further details on inverse interpolation see Traub [10] .

As before we begin by assuming that the minimum is bracketed by a and b . It becomes necessary to consider the two possible forms of $F(\alpha)$ separately. For simplicity we assume that $F(a) < F(b)$. The computation of $\hat{u}$ is then done as follows:

(a) $F=F$.

(i) All discontinuities located in $[a,b]$ are estimated by inverse interpolation and the estimates z_i are ordered and stored. If there is at least one discontinuity located between x and w (i.e. there is at least one $f_i(\alpha)$ with different sign at x and w) then the inverse interpolation is done at a and b since this is the most reliable choice, but otherwise each estimate is first made by inverse interpolation at the points x and w , and this is then replaced by the estimate using a and b only if the first estimate lies outside $[a,b]$. This is done because the good rates of convergence properties of successive inverse interpolation apply only if the best points are used for each fit. Note that although x and w are usually the two points with lowest values of $F(\alpha)$ it is clear that if $\overset{*}{\alpha}$ lies at a discontinuity with $f_k(\overset{*}{\alpha}) = 0$, then ultimately x and w will also be the two points with lowest value of $|f_k(\alpha)|$. If a and b are not equal to x and w , then ultimately there cannot be any discontinuity between x and w even if the minimum $\overset{*}{\alpha}$ is a discontinuity, and hence this strategy cannot impede the rate of convergence. It is possible that the inverse interpolation estimate using a and b lies outside [a,b]; if this happens, it is replaced by the secant estimate.

(ii) For convenience we set y_1 to the value a and y_2 to the smallest of the values $\{z_i\}$. The points y_1 and y_2 represent the current discontinuities as we examine them from left to right. We initially define the function $h(\alpha)$ by $F^{(a)}(\alpha)$ as defined by (3.4). The function $h(\alpha)$ is the differentiable function thought to coincide with $F(\alpha)$ between the discontinuities estimated by y_1 and y_2.

(iii) The point s is set to the estimate of the minimum of $h(\alpha)$ using (direct) cubic interpolation. As in the low overhead version, the cubic interpolation is done at the points a and b if there is at least one discontinuity located between x and w , and otherwise is done at the points x and w . If $y_1 \leq s \leq y_2$, then $\hat{u}$ is set to s . If s is undefined, which will be the case if h(o) is linear, then s is defined as either $+\infty$ or $-\infty$ by assuming h(o) is linear and comparing its values at x and w or a and b . For example, if $h(a) < h(b)$, then s is set to $-\infty$.

(iv) If $s < y_1$ then $\hat{u}$ is set to y_1 as then the differentiable functions coinciding with $F(\alpha)$ on either side of y_1 each appear to have their minimum on the opposite side of y_1 .

(v) If $s > y_2$ then y_1 is set to y_2 , y_2 is set to the next smallest of the values of the $\{z_i\}$, or to b if there are none greater than y_2 , $h(\alpha)$ is set to the function thought to coincide with $F(\alpha)$ in the new interval $[y_1, y_2]$, and steps (iii), (iv) and (v) are repeated. The new function $h(\alpha)$ is obtained by adding to the old $h(\alpha)$ the function $\pm f_k(\alpha)$, where z_k is the old value of y_2 and the sign is the sign of $f_k(b)$. However, if the old value of y_2 is b , the process is terminated with $\hat{u}$ set to b . This is illustrated in Figure 7 .

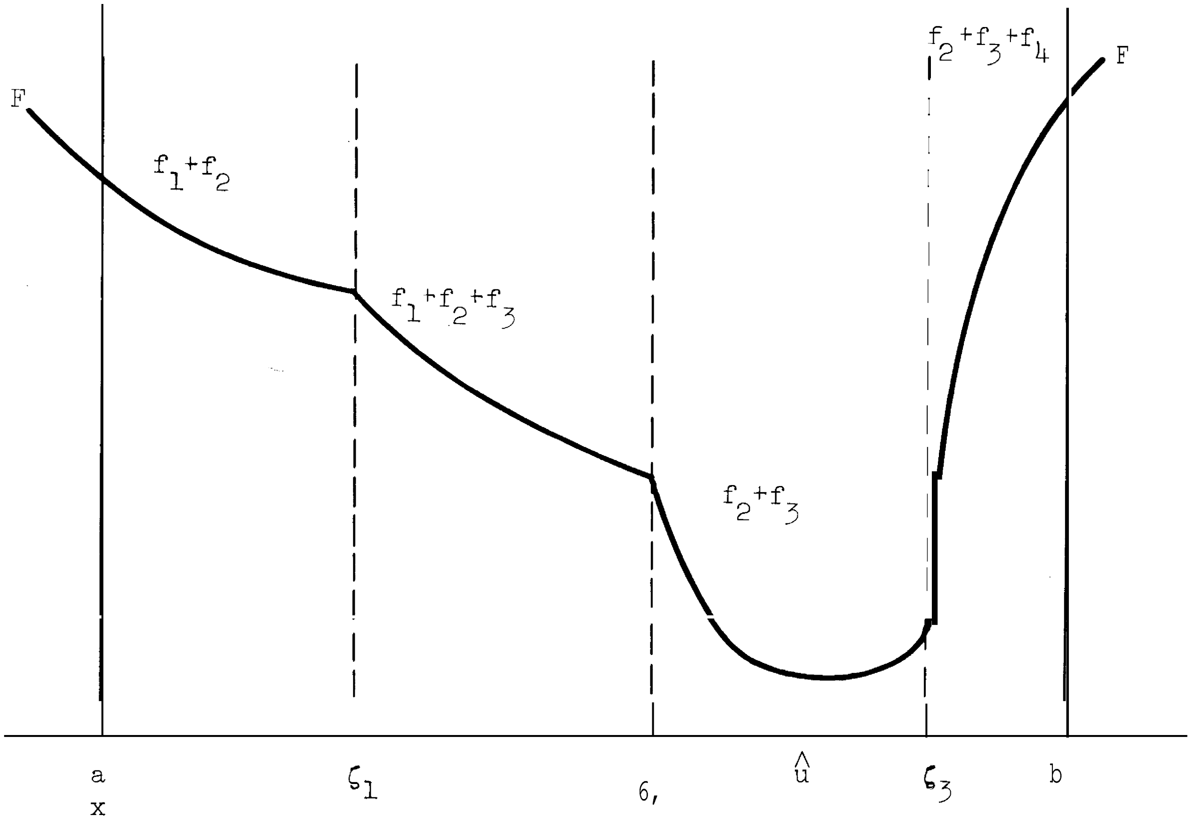
(b) $\underline{F = F_M}$.

(i) In this case in order to recognize the discontinuities it is necessary to estimate them in stages. We therefore begin by setting y_1 to a and k_1 to j(a) as defined by (2.2).

(ii) The point s is set to the estimate of the minimum of $f_{k_1}(\alpha)$ using (direct) cubic interpolation. As before, the interpolation is done at a and b if at least one discontinuity is located between x and w , i.e. if $j(x) \neq j(w)$, and otherwise it is done at x and w. Also if this makes s undefined it is set to $\pm\infty$ as before.

(iii) If $s < y_1$ then $\hat{u}$ is set to y_1 and the process is terminated.

(iv) The zeros of the functions $f_{k_1}(\alpha) - f_i(\alpha)$ for all $i \neq k_1$ are estimated by inverse interpolation. As in (ii> the points a and b are used if $j(x) \neq j(w)$ and otherwise x and w are used. Then y_2 is set to the minimum of those estimates which are greater than y_1 . If there are none greater than y_1 and less than b , then y_2 is set to b . Also k_2 is defined such that y_2 is the estimate of the zero of $f_{k_1}(\alpha) - f_{k_2}(\alpha)$, unless $y_2 = b$ when k_2 is undefined. If $s \leq y_2$,



Determining $\overset{A}{u}$ for $F = F_S$ in the higher overhead version. The zeros ζ_1 (of f_3), ζ_2 (of f_1) and ζ_3 (of f_4) are estimated first. Then the polynomial fits to each function are successively made. Provided that the minimum predicted by the fit to f_2+f_3 lies between the estimates of ζ_2 and ζ_3 , $\hat{u}$ is set to this point. The low overhead version would have set $\hat{u}$ to the average of estimates of ζ_1, ζ_2 and ζ_3 .

FIGURE 7

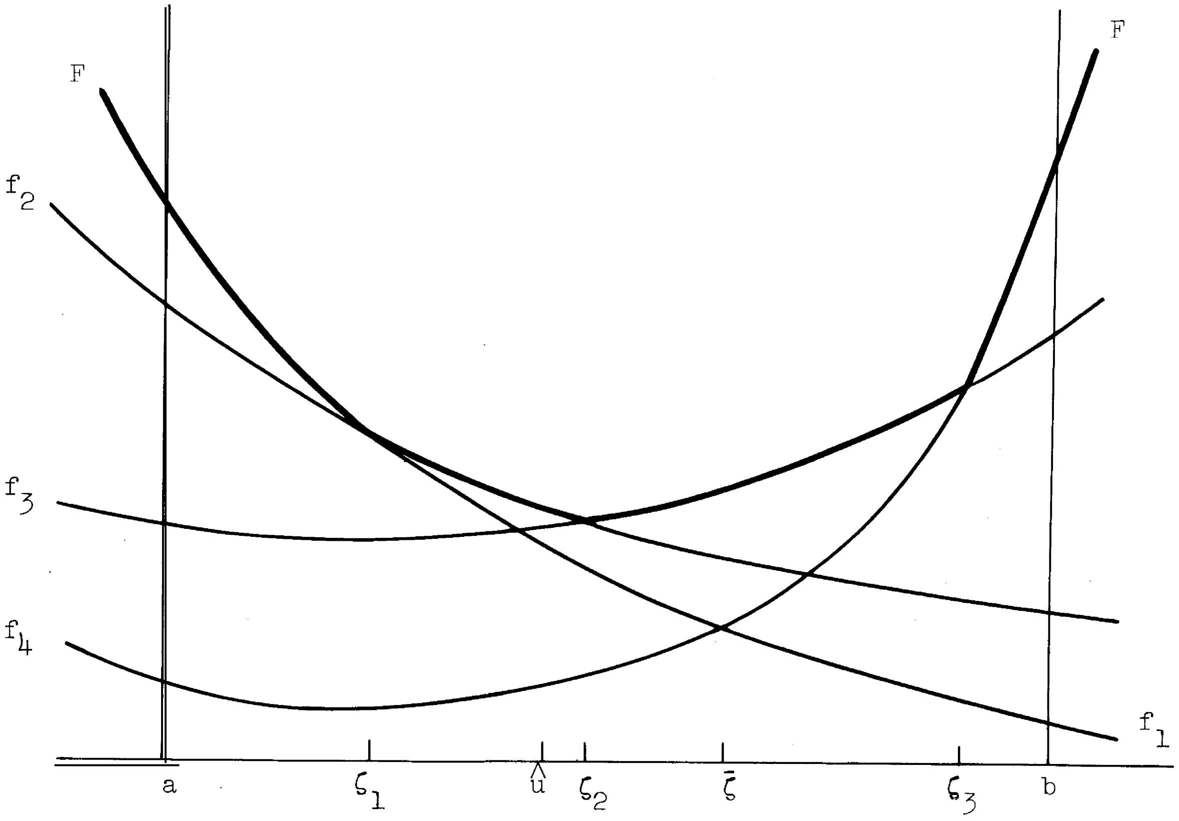
then $\hat{u}$ is set to s and the process is terminated.

(v) If $s > y_2$ then k_1 is set to k_2 , y_1 is set to y_2 , and steps (ii) to (v) are repeated, unless $y_2 = b$ when the process is terminated with $\hat{u}$ set to b . See Figure 8 for an illustration.

<u>However</u> if during the execution of the above k_2 is initially set to j(b) in part (iv) , i.e. we have $k_1 = j(a)$ and $k_2 = j(b)$ indicating only one discontinuity between a and b , <u>then in all subsequent computations of $\hat{u}$</u> part (iv) is replaced by

(iv-2) If $y_1 = a$ then y_2 is set to the estimate of the zero of $f_{j(a)} - f_{j(b)}$, using inverse interpolation either at a and b or at x and w , depending-on whether $j(x) = j(w)$ as before. Since we know there is a zero of this function in [a,b] , we replace the estimate using x and w ,if it lies outside [a,b] , by the estimate using a and b , and replace the estimate using a and b if necessary by the secant estimate. If $y_1 \neq a$, then y_2 is set to b . Note that estimating only the zero of $f_{j(a)} - f_{j(b)}$ is a safe strategy even though there may still be more than one discontinuity in [a,b] . An example of an unsafe strategy would be to estimate only the zero of, say, $f_2 - f_3$, if $\hat{u}$ had been set to.the estimate of the zero of this function several times already. Also note that the strategy may never be invoked since if the estimates converge to $\overset{*}{\alpha}$ from one side the interval of uncertainty may always contain more than one discontinuity. As in the low overhead version this is the only place where the definition of $\hat{u}$ depends on retaining any information from the previous iteration other than a,b,x,w and the function and derivative values.

Notice that for $F = F_M$ in part (iv) it would be *possible to exclude



Determining $\hat{u}$ for $F = F_M$ in the higher overhead version. The zeros ζ_1, ζ_2, 63 are successively estimated and the polynomial fits successively made to each function. Provided that the polynomial fit to f_2 is decreasing to the right of the estimate of ζ_2 and that to f_3 is decreasing to the left of it, $\hat{u}$ is set to the estimate of ζ_2. The low overhead version would have set $\hat{u}$ to an estimate of $\bar{\zeta}$.

FIGURE 8

from consideration those i such that $k_1 = i$ for an old value of k_1 or such that the estimate of the zero of $f_{k_1}(\alpha) - f_i(\alpha)$ lies outside $[a,b]$ for an old value of k_1.

It is worth noting that the choice of $\hat{u}$ in the higher overhead version requires of the order of $m \cdot \bar{m}$ operations in the case that $F = F_M$ (in part (iv)), where $\bar{m}$ is the number of discontinuities in the interval of uncertainty, but only of the order of m operations for $F = F_S$. If we were not permitted to store the $\{z_i\}$ in the case $F = F_S$, there would also be order $m \cdot \bar{m}$ operations required for this case. However, it does not appear possible to utilize storage in a similar way to reduce the operation count for $F = F_M$ since there are too many possible discontinuities to be stored in advance.

In both the above descriptions for $F = F_S$ and $F = F_M$ we have assumed that $F(a) < F(b)$ but clearly when this is not true the roles of a and b are simply interchanged and the discontinuities are examined from right to left instead of left to right.

As in the low overhead version we end the section by considering the case where $\overset{*}{\alpha}$ is not yet bracketed. The choice of $\hat{u}$ is made in much the same way as in the case that $F(a) < F(b)$, except that x and w are used for both the direct and inverse interpolations, and in the case $F = F_S$ the zeros of $f_i(\alpha)$ for all i must be estimated instead of just those thought to lie in $[a,b]$. Clearly instead of terminating if y_1 becomes b, the computation of $\hat{u}$ must terminate if y_1 becomes undefined and the safeguards will then choose a reasonable new u.

3.2.4 The case without derivatives.

We do not describe this in any detail but outline the changes to be made to the low and higher overhead version for computing $\hat{u}$. We now have the extra point v defined in Section 3.1. In the low overhead version the Newton steps to the discontinuities are replaced by secant steps. We use the points a and b when there are discontinuities located between w and v (and $\overset{*}{\alpha}$ is bracketed) and switch to using x and w when this is no longer true. In the higher overhead version the inverse cubic interpolation is replaced by inverse quadratic interpolation at the points a, x and b initially (if $\overset{*}{\alpha}$ is bracketed) and the points x, w and v ultimately. In both versions the (direct) cubic interpolation estimate of a minimum is replaced by quadratic interpolation, again at a,x, and b, or at x,w and v .

3.2.5 Convergence Results.

We now give the convergence results for the theoretical procedures associated with the algorithms described above. By theoretical procedures we mean exactly what was explained in Section 3.1.3.We assume that $F(\alpha)$ has one of the forms (2.1) and that an upper bound $\bar{b}$ on the interval of uncertainty is known.

Theorem 4. The theoretical procedures corresponding to both the low and higher overhead versions described above for both $F = F_S$ and $F = F_M$, in the cases with and without derivatives, all produce a sequence of points $\{u_k\}$ converging to a point $\overset{*}{\alpha}$ which is either a local minimum or generalized stationary inflection point of $F(\alpha)$ on $[0,\bar{b}]$. Furthermore if $f_i^{(1)}(\alpha)$ is Lipschitz continuous on [a,b] for $1 \le i \le m$, and $u_k \neq \overset{*}{\alpha}$ for all k , then we have the following. If either

(a) $F''(\overset{*}{\alpha})$ exists and is positive, and it is not true that $f_i(\overset{*}{\alpha}) = 0$ for some i if $F = F_S$, or that $f_i(\overset{*}{\alpha}) = f_k(\overset{*}{\alpha}) = F(\overset{*}{\alpha})$ for some ifk if $F = F_M$, or

(b) $F'(\overset{*}{\alpha}) \neq 0$, $F'_+(\overset{*}{\alpha}) \neq 0$, and $f_i(\overset{*}{\alpha}) = 0$ for exactly one i , with $f'_i(\overset{*}{\alpha}) \neq 0$, if $F = F_S$, or $f_i(\overset{*}{\alpha}) = f_k(\overset{*}{\alpha}) = F(\overset{*}{\alpha})$ for exactly one pair $i \neq k$, with $f'_i(\overset{*}{\alpha}) \neq f'_k(\overset{*}{\alpha})$,if $F = F_M$, then the asymptotic convergence rate is given by the following table.

Algorithm	Convergence rate	
	Case (a) above	Case (b) above
Low overhead with derivatives	2	2
Low overhead without derivatives	1.324...	1.618...
Higher overhead with derivatives	2	2.732...
Higher overhead without derivatives	1.324...	1.83%.

In order to prove this theorem we need several more lemmas in addition to those of Section 3.1.3. The first two are similar to Lemmas 3 and 4 , and, as before, we present the first without proof.

Lemma 5.

If $\overset{*}{u}$ is not a zero of a differentiable function $\varphi(\alpha)$ then $\exists\ \delta > 0$ s.t. the interval $(\overset{*}{u} - \delta , \overset{*}{u} + \delta)$ contains no zero of $\varphi(\alpha)$ and such that a secant or Newton step to the zero of $\varphi(\alpha)$ using a point inside the interval is good enough that the estimate of the zero lies outside the interval.

Lemma 6.

Assume each point in the subsequence $\{u_{j_k}\}$ is generated by either a Newton or secant step to the zero of $\varphi(\alpha)$ using the point x_{j_k}. If $x_{j_k} \to \overset{*}{u}$ and $u_{j_k} \to \overset{*}{u}$, then $\overset{*}{u}$ is a zero of $\varphi(\alpha)$.

Proof.

Identical to that for Lemma 4 using Lemma 5 instead of Lemma 3.

Lemma 7.

Assume the hypotheses of Lemma 4 except that the approximating polynomials are fitted to a continuously differentiable function $\Phi(\alpha)$ instead of $F(\alpha)$. Then if there is a sub-sequence of $(u_{j_k}\}$, namely $\{u_{i_k}\}$, s.t. $F(u_{i_k}) = \Phi(u_{i_k})$ $\forall$ k, the same result holds as for Lemma 4.

Proof.

By Lemma 4 $\overset{*}{u}$ must be a LM or GSIP of $\Phi(\alpha)$. Without loss of generality assume that a subsequence of $\{u_{j_k}\}$, namely $\{u_{i_k}\}$, converges to $\overset{*}{u}$ from the left. Since $\Phi(\alpha)$ is continuously differentiable we can write

$$\Phi'(\overset{*}{u}) = \lim_{k \to \infty} \frac{\Phi(u_{i_k}) - \Phi(\overset{*}{u})}{u_{i_k} - \overset{*}{u}} = F'(\overset{*}{u}) = 0.$$

The fact that $\overset{*}{u}$ must be a LM or GSIP of $F(\alpha)$ follows from this and the fact that $F(u_{i_k}) \to F(\overset{*}{u})$ from above.

Proof of Theorem 4.

We restrict our attention to the low overhead version. Either there is a subsequence $\{u_{j_k}\}$ with $u_{j_k} = \bar{u}_{j_k}$ for all k , or there exists K s.t. $u_k = \hat{u}_k$ for $k \geq K$. In the former case as before we obtain the first part of the result from Lemma 1. Therefore assume the latter case. There must be a subsequence $(u_{j_k}\}$ either (i) consisting entirely of points generated by polynomial approximation fitting to a certain differentiable function $\Phi(\alpha)$, or else (ii) consisting entirely of points generated by Newton or secant steps to the zero of a differentiable function $\varphi(\alpha)$.

Case (i).

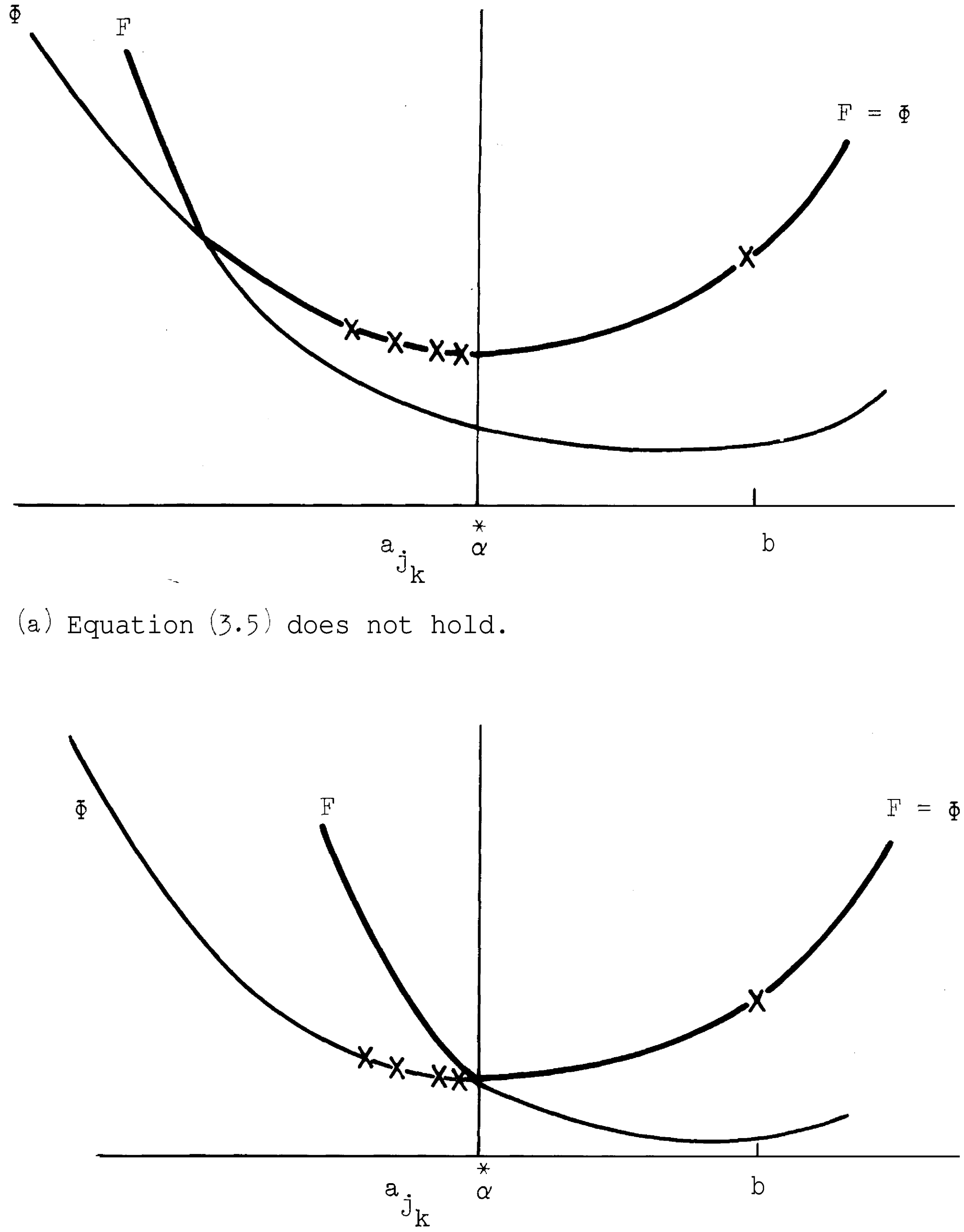
By Lemmas 2 and 4 , $u_{j_k} \to \overset{*}{u}$ with $\overset{*}{u}$ a LM or GSIP of $\Phi(\alpha)$. (Note that the quadratic approximation always uses the two best points for two of its three points and similarly the cubic approximation always uses the best point). Suppose there is a subsequence $(u_{i_k}\}$ of $\{u_{j_k}\}$ such that $F(u_{i_k}) = \Phi(u_{i_k})$ for all k . Then by Lemma 7 $\overset{*}{u}$ is a LM or GSIP of $F(\alpha)$. Otherwise $\exists$ K s.t.

$$F(u_{j_k}) \neq \Phi(u_{j_k}) \text{ for } k \geq K . \qquad (3.5)$$

Since $x_{j_k} \rightarrow \overset{*}{u}$ and $w_{j_k} \rightarrow \overset{*}{u}$ we know either $a_{j_k} \rightarrow \overset{*}{u}$ or $b_{j_k} \rightarrow \overset{*}{u}$. Without loss of generality assume the former, i.e. $a_{j_k} \rightarrow \overset{*}{u}$. Because of the way the fit is chosen in the low overhead version, we have for each k either $F(a_{j_k}) = \Phi(a_{j_k})$ or $F(b_{j_k}) = \Phi(b_{j_k})$. By taking subsequences (but not writing them explicitly), we can assume that either $F(a_{j_k}) = \Phi(a_{j_k})$ for all $k \geq K$, or $F(b_{j_k}) = \Phi(b_{j_k})$ for all $k \geq K$. The former contradicts (3.5) , as we could write $(a_{j_k}\}$ as a subsequence of $\{u_{j_k}\}$ converging to $\overset{*}{u}$, so we assume the latter. Since $\Phi(\alpha)$ and $F(\alpha)$ agree at b_{j_k} but not at u_{j_k} , there must exist a discontinuity ζ such that $u_{j_k} \leq \zeta \leq b_{j_k}$ for all $k > K$, and hence $\overset{*}{u} \leq \zeta$. Now consider the estimate of ζ at each step, namely $\bar{z}_{j_k}$, which results from a Newton or secant step using the best point x_{j_k} . We have $a_{j_k} \leq \bar{z}_{j_k} \leq b_{j_k}$. On the other hand, $\bar{z}_{j_k}$ must lie outside $[u_{j_k} , b_{j_k}]$ or the step to u_{j_k} would not be accepted as it crosses a discontinuity estimate. Thus $a_{j_k} \leq \bar{z}_{j_k} \leq u_{j_k}$ and $\bar{z}_{j_k} \rightarrow \overset{*}{u}$. By Lemma 5 this is not possible unless $\zeta = \overset{*}{u}$. We therefore have that $\Phi(\alpha)$ agrees with $F(\alpha)$ on an interval to the right of and including $\overset{*}{u}$, and hence that $F'_+(\overset{*}{u}) = 0$ as $\Phi'(\overset{*}{u}) = 0$. Since $a_{j_k} \rightarrow \overset{*}{u}$, we know $F(\alpha)$ is non-decreasing on the left of $\overset{*}{u}$, so $\overset{*}{u}$ is a LM or GSIP of $F(\alpha)$ (see Figure 9).

Case (ii).

We have $\{u_{j_k}\}$ where each point is generated by a Newton or secant step to the zero of a differentiable function $\varphi(\alpha)$. By Lemmas 2 and 5 ,



(a) Equation (3.5) does not hold.

(b) Equation (3.5) holds.

Two possibilities in case (i) of the proof of Theorem 4 .

FIGURE 9

$u_{j_k} \to \overset{*}{u}$ and $\overset{*}{u}$ must be a zero of $\varphi(\alpha)$. Suppose $\overset{*}{u}$ is not a LM or GSIP, and without loss of generality assume $F'_{+}(\overset{*}{u}) < 0$. Let $\Phi(\alpha)$ be the differentiable function coinciding with $F(\alpha)$ on the right side of $\overset{*}{u}$ The algorithm will not permit u_{j_k} to be set to the estimate z of the zero of $\varphi(\alpha)$ if a point to the right of" z is produced by polynomial approximation to the function thought to coincide with $F(\alpha)$ to the right of z . In the low overhead version the function in question is $F^{(b)}(\alpha)$, and since u_{j_k} is then always set to $\bar{z}_{j_k}$ ultimately we must have $\bar{z}_{j_k}$ equal to a specific discontinuity estimate and $F^{(b)}(\alpha) \equiv \Phi(\alpha)$. (In the higher overhead case the function in question is clearly ultimately equal to $\Phi(\alpha)$.) Thus by Lemma 3 this fit is ultimately good enough that a_{j_k} cannot be set to the estimate of the zero of $\varphi(\alpha)$, which is a contradiction. Hence $\overset{*}{u}$ must be a LM or GSIP.

This completes the first part of the proof for the low overhead version. We omit the proof for the higher overhead version since it is similar. The main difference is the replacement of the Newton and secant results by analogous ones for successive inverse interpolation.

The hypotheses of the second part of the theorem ensure that ultimately the points u_k are generated entirely by successive estimates of a minimum using (direct) cubic or quadratic interpolation or entirely by successive estimates of a zero using the secant method, Newton's method, or inverse cubic or quadratic interpolation. They also ensure that ultimately the best points are used for the interpolation and hence that the rate of convergence is not impeded. The convergence rates for successive cubic or quadratic interpolation were quoted in Section 3.1.3, those for the secant method and Newton's method are well known, and those for inverse interpolation may be found in Traub [10,p.66]. This completes the proof of the theorem.

Finally we note that as in the differentiable case it does not make sense to talk about just any local minimum of the computed function $\hat{F}(\alpha)$. It is easy to verify that Theorems 2 and 3 hold for the nondifferentiable algorithms as well, where $\hat{F}'(\alpha)$ is the quantity resulting from computing $F'(\alpha)$ as defined in (2.2).

4. Steplength Algorithms.

In this section we discuss how to choose $\alpha^{(k)}$ (see Section 1), when minimizing an n dimensional function of the type given by (1.1). In order to prove convergence for descent methods the steplength has to meet certain criteria. The function must be "sufficiently decreased" with respect to the steplength, and the steplength must not be too small (see Ortega and Rheinboldt [2,p.490]). For differentiable functions a typical criterion to ensure that the first condition is satisfied is

$$F^{(k)} - F(x^{(k)} + \alpha^{(k)}p) \geq - \mu \alpha^{(k)} g^{(k)^T} p \qquad (4.1)$$

where μ is a preassigned scalar, $0 < \mu < 1$. (We have now omitted the superscript from $p^{(k)}$, and have denoted $F(x^{(k)})$ and $g(x^{(k)})$ by $F^{(k)}$ and $g^{(k)}$.) As was mentioned in the introduction, such criteria do not in general define a unique point. Many elementary algorithms have been proposed which satisfy them. However it is important to realize that for a practical algorithm mere convergence in the limit is only of academic interest. We are interested in the finite sequence $\{x^{(k)}\}$, k=1,...N, where N is preferably small, and where $x^{(N)}$ is "close" to $\overset{*}{x}$. The greater the reduction in $F(x)$ per iteration usually the lower the value of N . It is necessary however to limit the effort expended on determining

$\alpha^{(k)}$ since this in itself could be an infinite process.

For algorithms which are designed to minimize differentiable functions Gill and Murray [1] proposed choosing the steplength by proceeding to compute a local minimum of $F(x^{(k)} + \alpha p)$ using the algorithm described in Section 3.1 and terminating this prematurely (possibly after a single function evaluation). When derivatives are available the termination condition is

$$F(x^{(k)} + \alpha p) < F^{(k)}$$

and

$$|g(x^{(k)} + \alpha p)^T p| < - \eta g^{(k)^T} p \tag{4.2}$$

where η is a preassigned scalar, $0 \le \eta \le 1$. A check is then made as to whether this step satisfies (4.1) with μ set to a small value such as 10^{-4}. The experience with such a procedure in the many cases that were checked is that the resulting step always satisfied (4.1). If (4.1) is not satisfied, the step is successively contracted by a factor of one half until it satisfies (4.1). It is proved in [1] that this strategy is sufficient to ensure the overall convergence.

Clearly the smaller the parameter η is, the greater the reduction obtained in $F(x)$ but the more evaluations of $F(x)$ required. The optimal choice of η will vary both with the algorithm within which the procedure is incorporated and the problem being solved. Fortunately for a particular algorithm a near optimal value of η can be predetermined That different algorithms will require different choices of η arises from the relative effort of computing the search direction p and performing additional iterations of the univariate search (recall that the more accurate the univariate search, the fewer iterations of the multidimensional algorithm required). If for example in a Newton-type method it was expensive to evaluate the Hessian matrix (required to compute p only) compared to evaluating $F(x)$ and $g(x)$ then a small value of η would be warranted. Similarly if n was large making the housekeeping operations of obtaining p significant, then again a small value of η would be warranted. For most algorithms, however, the optimal η under most circumstances is in the range 0.5 - 0.9 .

The termination criterion (4.2) is clearly inappropriate for nondifferentiable functions since if $\overset{*}{\alpha}$ is a discontinuity there may be no value of α which satisfies it. To achieve a similar objective for nondifferentiable functions we propose the following. Let $\bar{\alpha}$ be the first point in the sequence generated by one of the minimization algorithms with derivativesdescribed in Section 3.2 such that

$$F(x^{(k)}+ \bar{\alpha} p) < F^{(k)}$$

and either

$$|g(x^{(k)}+ \bar{\alpha}\ p)^{T} p| \leq\ - \eta\ g^{(k)^{T}} p \qquad (4.3a)$$

or

$$|\varphi(x^{(k)}+ \bar{\alpha}\ p)| \leq \eta\ |\varphi_0| \qquad (4.3b)$$

The test (4.3b) is done <u>only</u> if the generation of the next point <u>after</u> $\bar{\alpha}$ in the univariate search entails setting the new point u to an estimate of a specific discontinuity, namely the zero of the function $\varphi(x^k + \alpha\ p)$. The scalar φ_0 is the value of $\varphi(\alpha)$ at the first point at which we assume $\overset{*}{\alpha}$ to be the zero of $\varphi(\alpha)$. The required steplength $\alpha^{(k)}$ is set to $\bar{\alpha}$ provided this satisfies any criteria such as (4.1) required for guaranteed convergence of the algorithm. In the unlikely event $\bar{\alpha}$ is not satisfactory $\alpha^{(k)}$ is set to the first satisfactory member of the sequence $\{(\frac{1}{2})^i \bar{\alpha} | \ i = 1,2....\}$.

In the case without derivatives, $g(x^{(k)} + \bar{\alpha} p)^T p$ in (4.3a) is replaced by

$$\frac{F(x^{(k)} + \sigma p) - F(x^{(k)} + \bar{\alpha} p)}{\sigma - \bar{\alpha}} ,$$

where σ is the last point in the sequence obtained in the univariate search which is less than $\bar{\alpha}$. We assume that an estimate of $g^{(k)}$ is available as a result of determining p .

5. Extensions to a Wider Class of Functions.

Although we have confined our attention so far to functions of the type (1.1), the algorithms presented here can be extended to handle a wider class of functions. Two common types of nondifferentiable functions are those arising from the ℓ_1 and ℓ_∞ approximation problems, namely

$$F_1(x) = \sum_{i=1}^{m} |f_i(x)| \qquad \text{and} \qquad F_\infty(x) = \max_{1 \le i \le m} |f_i(x)| .$$

A third is the class of nondifferentiable penalty functions arising from general minimization problems subject to inequality or equality constraints (see Conn and Pietrzykowski [11] and Han [12]):

$$F_P(x) = f_0(x) + \rho_1 \sum_{i=1}^{m_1} \max(0, f_i(x)) + \rho_2 \sum_{i=m_1+1}^{m} |f_i(x)| .$$

The functions $F_1(x)$ and $F_\infty(x)$ could be transformed to the type (1.1) , as could $F_P(x)$ if a suitable positive term were added to $f_0(x)$. However, to do so is both artificial and unnecessary, and although the performance of the steplength algorithms would be satisfactory, the transformation would be likely to introduce degeneracy into the n dimensional algorithm. It is therefore much more satisfactory to consider the following two types of functions:

$$F_{GS}(x) = \sum_{i=1}^{m} \sigma(\theta_i, f_i(x))$$

where $\sigma(\theta_i, f_i(x)$ is one of $|f_i(x)|$, $\max(0,f_i(x))$, $f_i(x)$, $\min(0,f_i(x))$, $-|f_i(x)|$, according to the value of θ_i, and (5.1)

$$F_{GM}(x) = \max\left(\max_{1 \leq i \leq m_1} |f_i(x)|, \max_{m_1+1 \leq i \leq m} f_i(x)\right).$$

Note that there are functions of the type $F_{GS}(x)$ which cannot be transformed to the type (1.1). These two types of functions clearly have their discontinuities defined in a similar way to that described for the functions F_S and F_M, and it is easy to modify the algorithms to cope with these more general cases. Since the modifications introduce little additional overhead, our implementations of the algorithms cope with these wider classes of functions.

As indicated in Section 1, the ideas of these algorithms could be extended to handle virtually any continuous function whose directional derivatives exist everywhere and whose discontinuities are given by the roots of known differentiable functions. We believe however that most such functions arising in practice are either of the type (5.1) or else could easily be transformed to this type.

Finally we note that it would be possible to extend the algorithms described above for use in minimizing certain differentiable functions with discontinuities in the second derivative. If the minimum is at a point of discontinuity in the second derivative, the convergence rate will normally be only linear for the differentiable case algorithm described in [1], but if the minimum is also a root of a differentiable function, the good rates of convergence for the nondifferentiable algorithms could be achieved if the algorithms were extended properly. An example of such a function would be $F(x) = x^2$ on $[0,\infty)$ and $F(x) = \frac{1}{2} x^2$ on $(-\infty,0]$.

6. Implementation and Numerical Results.

The algorithms described in this paper have been implemented in Fortran. They make use of the computer programs for the algorithms described in [1], which are documented in [13] and form part of the Numerical Optimization Software Library at the National Physical Laboratory. Hence the safeguards are attended to by the existing programs and the new programs essentially compute $\hat{u}$ at each iteration and include the extra steplength termination criterion.

We present the results of some test runs of the higher overhead steplength algorithm for $F = F_S$ using derivatives, and compare them with running the algorithm of [1], intended for differentiable functions, on the same function. Although we have not yet had extensive numerical experience with the new algorithms, the results illustrate their potential advantages. The univariate function is

$$F(x) = f_1(x) + \max(f_2(x),0) + \max(f_3(x),0)$$

where

$$f_1(x) = -\cos x$$

$f_2(x) = 4(x-1)$ and either

(a) $f_3(x) = -10 \sin(0.5(x-0.1))$ or

(b) $f_3(x) = -10 \sin(0.5(x+0.1))$.

The initial point is $x_0 = -1.2$, the direction of search is $p = 1$, and the initial step is $\alpha_0 = 1$. In case (a) the points of derivative discontinuity are $x = 0.1$ and $x = 1.0$, and the minimum is at the first of these. In case (b) the points of discontinuity are $x = -0.1$ and $x = 1.0$ and the minimum is at $x = 0.0$, where the function is differentiable. Results are given for several values of η : $\eta = 10^{-6}$ for an "exact" line search, and $\eta = 0.1$ and 0.5 for "slack" searches. The tolerances ϵ and τ are set to 10^{-6} . The results were obtained on an IBM 370/168 using double precision, i.e. approximately 14 decimal digits of accuracy. They appear in Table 1. The number of function evaluations includes the evaluation $F(x_0 + \alpha_0 p)$.

The results illustrate that as well as being far more efficient than the algorithm of [1] for an exact line search where the minimum is at a discontinuity, the new algorithm can also be significantly more efficient for slack line searches where the minimum may or may not be at a discontinuity. In all cases for large η the new algorithm required less function evaluations and in all but one also produced a lower point. In case (a), for $\eta = 0.5$, the algorithm terminated with $x = x_0 + \alpha_0$, since it determined that the best step to take next was to the zero of $f_3(x)$ but that the step from x_0 to $x_0 + \alpha_0$ had already achieved a reduction in $|f_3(x)|$ sufficiently large enough to allow it to stop. Note that in case (a) if the left and right derivatives at the solution had been sufficiently higher, the algorithm of [1] would have been unable to terminate until the length of the interval of uncertainty was reduced to $2\ \mathrm{tol}(x)$ even for large η , since it would be unable to reduce the gradient to $\eta \cdot F'(x_0)$.

		Results using the new algorithm			Results using the algorithm of [1].		
	h	Number of function evaluations	Final value of x	Final value of F(x)	Number of function evaluations	Final value of x	Final value of F(x)
case (a)	10^{-6}	3	0.100	- 0.995	30	0.100	- 0.995
	0.1	2	0.103	- 0.995	4	0.442	- 0.904
	0.5	1	- 0.200	- 0.514	3	0.833	- 0.672
case (b)	10^{-6}	4	0.0	- 1.0	7	0.0	- 1.0
	0.1	2	- 0.020	- 0.998	4	0.183	- 0.983
	0.5	2	- 0.020	- 0.998	3	0.771	- 0.717

TABLE I

7. Concluding Remarks.

A fundamental part of algorithms and software for minimizing differentiable functions of several variables is an efficient steplength algorithm. The basic algorithms described by Gill and Murray [1] have been incorporated in the implementation of more than 50 different routines for unconstrained and constrained optimization. We believe that the same potential exists for developing software for nondifferentiable functions. Although there is not as yet the same variety of routines for this class of problems, the existence of a powerful steplength algorithm will in itself provide a stimulus. The routines should also prove useful when nondifferentiable functions are used as merit functions for solving constrained optimization problems.

Acknowledgements.

The authors would like to thank Dr. Philip E. Gill for help with numerous points in preparing both the manuscript and the computer programs. The second author would also like to thank the staff of the Division of Numerical Analysis and Computer Science at the National Physical Laboratory for their hospitality during his stay there and also to thank Professor Gene H. Golub for the opportunity to visit there while being supported at Stanford University.

References.

[1] Gill, P.E. and Murray, W., Safeguarded steplength algorithms for optimization using descent methods, National Physical Laboratory Report NAC 37 (1974).

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

[3] Brent, R.P., Algorithms for minimization without derivatives, Prentice-Hall, Englewood Cliffs, N.J. (1973).

[4] Jarratt, P., An iterative method for locating turning points, Comp. J. 10 (1967) 82-84.

[5] Kowalik, J. and Osborne, M.R., Methods for unconstrained optimization problems, Elsevier, New York (1968).

[6] Overton, M.L., Rate of convergence of Davidon's algorithm for finding-the minimum of a function of one variable using successive cubic Hermite interpolation, manuscript, Computer Science Dept., Stanford University (1977).

[7] MACSYMA Reference Manual, The Mathlab Group, Laboratory for Computer Science, Massachusetts Institute of Technology.

[8] Bjorstad, P. and Nocedal, J., Analysis of a new algorithm for one-dimensional minimization, Computer Science Dept. Report STAN-CS-78-664, Stanford University (1978).

[9] Charalambous, C. and Conn, A.R., An efficient method to solve the minimax problem directly, SIAM J. Numer. Anal. 15 (1978) 162-187.

[10] Traub, J.F., Iterative methods for the solution of equations, Prentice-Hall, Englewood Cliffs, N.J. (1964).

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

[12] Han, S.P., A globally convergent method for nonlinear programming, JOTA 22 (1977) 297-309.

[13] Gill, P.E. et al., Documents for Subroutines LINDER and LINDIF, NPL Algorithms Library Reference Nos. E4/15/0 and E4/16/0, National Physical Laboratory (1976).