THE COMPLEXITY OF OBTAINING STARTING POINTS FOR SOLVING OPERATOR EQUATIONS BY NEWTON'S METHOD

H. T. Kung
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213

October, 1975

This research was supported in part by the National Science Foundation under Grant GJ32111 and the Office of Naval Research under Contract N0014-67-A-0314-0010, NR 044-422.

This report is to appear in Analytic Computational Complexity, Academic Press, 1975.

ABSTRACT

The complexity of searching for good starting points for iterations is studied. Global and non-asymptotic results are obtained: A useful lemma for proving lower bounds is identified, several optimal results are given for scalar equations, and upper bounds for operator equations are established by a new procedure for obtaining starting points for Newton's method.

1. INTRODUCTION

Algorithms for finding a root α of a nonlinear equation $f(x) = 0$ usually consist of two phases:

1. Search phase: Search for initial approximation(s) to α.
2. Iteration phase: Perform an iteration starting from the initial approximation(s) obtained in the search phase.

Most results in analytic computational complexity assume that good initial approximations are available and deal with the iteration phase only. Since the complexity, i.e., the time, of the computation for solving $f(x) = 0$ is really the sum of the complexities of both the search and iteration phases, we propose to study both phases. Moreover, we observe that the complexities of the two phases are closely related. The speed of convergence of the iteration at the iteration phase in general depends upon the initial approximation(s) obtained in the search phase. If we spend much time in the search phase so that "good" initial approximation(s) are obtained, then we may expect to reduce the time needed in the

iteration phase. This observation will be made precise in this paper. On the other hand, if we do not spend much time in the search phase and initial approximation(s) obtained are not so "good", then the complexity of the iteration phase could be extremely large, even if the corresponding iteration still converges. Some good examples of the phenomenon can be found in Traub and Woźniakowski [75]. All these show that the complexity of the iteration phase depends upon that of the search phase. Hence we feel that it is necessary to include both phases in the complexity analysis. Through this approach we can also obtain the optimal decision on when the search phase should be switched to the iteration phase, since it can be found by minimizing the total complexity of the two phases.

In this paper, we shall assume that f satisfies some property (or conditions), and include in our analysis the time needed in both the search phase and iteration phase. Note that it is necessary to assume f satisfies some property, since we have to make sure at least that there exists a root in the region to be searched. The general question we ask in the paper is how fast we can solve $f(x) = 0$ in the worst case, when f satisfies certain conditions.

In the following section we give the methodology to be used in the paper, which does not have the usual assumption that "good initial approximations are available". Instead, we assume that some property of the function f is known, i.e., f satisfies certain conditions. A useful lemma for proving lower bounds on complexity, in our methodology, is also given in the section.

Section 3 gives several relatively simple results for $f: R \to R$. The main purpose of the section is to illustrate the techniques for proving lower bounds. One of the results

shows that even if we know that $M \geq f'(x) \geq m > 0$ on an interval $[a,b]$ and $f(a)f(b) < 0$, it is impossible to solve $f(x) = 0$ by a superlinearly convergent method. However, if in addition $|f''|$ is known to be bounded by a constant on $[a,b]$, then the problem can be solved superlinearly.

In Section 4 we give upper bounds on the complexity for solving certain operator equations $f(x) = 0$, where f maps from Banach spaces to Banach spaces (Theorem 4.3). This section contains the main results of the paper. A procedure (Algorithm 4.2) is given for finding points in the region of convergence of Newton's method, for f satisfying certain natural conditions. The complexity of the procedure is estimated a priori (Theorem 4.2), and the optimal branching condition on when the search phase is switched to the iteration phase is also given. We believe that the idea of the procedure can be applied to other iterative methods for f satisfying various conditions. By a preliminary version (Algorithm 4.1) of the procedure, we also establish an existence theorem (Theorem 4.1) in Section 4.

Summary and conclusions of the paper are given in the last section.

2. METHODOLOGY AND A USEFUL LEMMA FOR PROVING LOWER BOUNDS

Let φ be an algorithm for finding a root α of $f(x) = 0$ and x the approximation to α computed by φ. Denote the error of the approximation x by

$$\delta(\varphi,f) = \|x-\alpha\|,$$

where $\|\cdot\|$ is a suitable norm. Consider the problem of solving $f(x) = 0$ where the function (or operator) f satisfies some property (or some conditions). Since algorithms based on the

property cannot distinguish individual functions in the class F of all functions satisfying the property, we really deal with the class F instead of individual functions in F. Define

$$\Delta_i = \inf_{\varphi \in \Omega_i} \sup_{f \in F} \delta(\varphi, f),$$

where Ω_i is the class of all algorithms using i units of time. Then the time t needed to approximate a root to within $\varepsilon > 0$ is the smallest i such that $\Delta_i \le \varepsilon$, and an algorithm $\varphi \in \Omega_t$ is said to be optimal for approximating a root to within $\varepsilon > 0$ if

$$\sup_{f \in F} \delta(\varphi, f) = \Delta_t.$$

Hence the complexity of the problem is determined by the sequence $\{\Delta_i\}$. We say the problem is solvable if $\{\Delta_i\}$ converges to zero, otherwise it is unsolvable. We are interested in solvable problems. For understanding the asymptotic behavior of the sequence $\{\Delta_i\}$, we study the order of convergence of $\{\Delta_i\}$, which is defined to be

$$p = \lim_{i \to \infty} (|\log \Delta_i|)^{1/i},$$

provided the limit exists. If $|\log \Delta_i|$ increases exponentially as $i \to \infty$, i.e., $p > 1$, we say the problem can be solved superlinearly. Our goal is to establish upper and lower bounds on Δ_i for given problems.

Upper bounds on Δ_i are established by algorithms. The following lemma is useful for proving lower bounds on Δ_i. The idea of the lemma has been used by many people, including

Brent, Wolfe and Winograd [73], Winograd [75], Woźniakowski [74], etc. under various settings. It is perhaps the most powerful idea so far for establishing lower bounds in analytic computational complexity.

Lemma 2.1. If for any algorithm using i units to time, there exist functions f_1, f_2 in F such that

(2.1) the algorithm cannot distinguish f_1 and f_2, and

(2.2) the minimum distance between any zero of f_1 and any zero of f_2 is $\geq 2\epsilon$,

then

$$\Delta_i \geq \epsilon.$$

Proof. Consider any algorithm using i units of time. Suppose that f_1, f_2 satisfy (2.1) and (2.2). Let α_1, α_2 be the zeros of f_1, f_2 respectively. By (2.1), the algorithm computes the same approximate x for f_1 and f_2. By (2.2)

$$|x-\alpha_1| + |x-\alpha_2| \geq |\alpha_1-\alpha_2| \geq 2\epsilon.$$

Hence either $|x-\alpha_1| \geq \epsilon$ or $|x-\alpha_2| \geq \epsilon$. ■

3. SOME RESULTS ON REAL VALUED FUNCTIONS OF ONE VARIABLE

In this section we shall give several relatively easy results to illustrate the concepts given in the preceding section, and the use of Lemma 2.1. We consider $f:[a,b]\subset R\to R$. For simplicity we assume that each function or derivative evaluation takes one unit of time and the time needed for other operations can be ignored.

Theorem 3.1. If f: [a,b] → R satisfies the following properties:

(3.1) f is continuous on [a,b], and

(3.2) $f(a) \leq 0$, $f(b) > 0$,

then $\Delta_i = (b-a)/2^{i+1}$.

Proof. It is clear that by binary search we have that $\Delta_i \leq (b-a)/2^{i+1}$. Let φ be any algorithm using i evaluations. Algorithm 3.1 below constructs f_1, f_2 such that (2.1) and (2.2) hold for $\epsilon = [(b-a)/2^{i+1}] - \delta$, and (3.1), (3.2) hold for $f = f_1$ or $f = f_2$, where $0 < \delta < [(b-a)/2^{i+1}]$. We first define

$$u(x) = 1,$$
$$v(x) = -1$$

for $x \in [a,b]$, and assume the first evaluation is at x_0.

Algorithm 3.1.

1. Set $\ell \leftarrow a$, $r \leftarrow b$, $m \leftarrow x_0$, $c(x) = u(x)$ for $x \in (a,b]$ and $c(a) = v(a)$.

2. If $m \notin [\ell,r]$, go to step 4.

3. If $m-\ell \geq r-m$, set $r \leftarrow m$. Otherwise, define $c(x) = v(x)$ for $x \in [\ell,m]$ and set $\ell \leftarrow m$.

4. Apply algorithm φ to function c(x) and compute the next approximation.

5. If algorithm φ has not terminated, set m to be the point where the next evaluation takes place and go to step 2.

6. Define f_1, f_2 by $f_1(x) = f_2(x) = c(x)$ for $x \in [a, \ell] \cup [r,b]$,

$$f_1(x) = \begin{cases} u(x) & \text{for } x \in [\ell+\delta, r), \\ \frac{x-\ell}{\delta}[u(\ell+\delta) - v(\ell)] + v(\ell) & \text{for } x \in (\ell, \ell+\delta), \end{cases}$$

and

$$f_2(x) = \begin{cases} v(x) & \text{for } x \in (\ell, r-\delta], \\ \frac{x-r}{\delta}[u(r) - v(r-\delta)] + u(r) & \text{for } x \in (r-\delta, r). \end{cases}$$

It is straightforward to check that $r-\ell \geq (b-a)/2^i$ and that the distance between any zero of f_1 and any zero of f_2 is $\geq r-\ell-2\delta$. Hence (2.2) is satisfied for $\epsilon = [(b-a)/2^{i+1}] - \delta$. It is also easy to see that (2.1), (3.1) and (3.2) hold for f_1, f_2. Hence by Lemma 2.1, we have $\Delta_i \geq [(b-a)/2^{i+1}] - \delta$. Since δ can be chosen arbitrarily small, we have shown $\Delta_i \geq (b-a)/2^{i+1}$. ■

Theorem 3.1 establishes that binary search is optimal for finding a zero of f satisfying (3.1) and (3.2). The result is well-known. The theorem is included here because its proof is instructive. By slightly modifying the proof of Theorem 3.1, we obtain the following result:

Theorem 3.2. If f: $[a,b] \to R$ satisfies the following properties:

(3.3) $f'(x) \geq m > 0$ for all $x \in [a,b]$, and
$f(a) < 0$, $f(b) > 0$,

then $\Delta_i = (b-a)/2^{i+1}$.

Proof. The proof is the same as that of Theorem 3.1, except that the functions u, v are now defined as

$$u(x) = m(x-a),$$
$$v(x) = m(x-b),$$

and the functions f_1, f_2 have to be smoothed so that they satisfy (3.3). ■

One can similarly prove the following two theorems.

Theorem 3.3. If $f: [a,b] \to R$ satisfies the following properties:

$f'(x) \leq M$ for all $x \in [a,b]$, and

$f(a) < 0$, $f(b) > 0$,

then $\Delta_i = (b-a)/2^{i+1}$.

By Theorems 3.2 and 3.3, we know that even if f' is bounded above or bounded below, we still cannot do better than binary search in the worst case sense.

Theorem 3.4. If $f: [a,b] \to R$ satisfies the following properties:

$M \geq f'(x) \geq m > 0$ for all $x \in [a,b]$, and

$f(a) < 0$, $f(b) > 0$

then $\Delta_i \geq (b-a)[(1-\frac{m}{M})^2/2]^{i+1}$.

Under the conditions of Theorem 3.4, Micchelli and Miranker [75] showed that

$$\Delta_i \leq \frac{1}{2}(b-a)(1-\frac{m}{M})^{\frac{i}{2}}.$$

Hence their algorithm is better than binary search when $\frac{m}{M} \geq \frac{3}{4}$. However, by Theorem 3.4, we know that the problem cannot be solved superlinearly, even when f' is known to be

bounded above and below by some constants. In order to assure that the problem can be solved superlinearly we have to make further assumptions on the function f. A natural way is to assume that $|f''|$ is bounded. This leads to the following

Theorem 3.5. If the conditions of Theorem 3.4 are satisfied and $|f''| \leq K$ on $[a,b]$, then the problem of finding a root of $f(x) = 0$ can be solved superlinearly.

Proof. We can use binary search to find a point x_0 which satisfies the conditions of the Newton-Kantorovich Theorem (see the next section for the statement of the theorem). It is easy to see that only a finitely many steps of binary search are needed to find x_0. Starting from x_0 the Newton iterates converge to a root superlinearly. ■

It should be noted that the binary search used in the above proof would not make sense for operators mapping from Banach spaces to Banach spaces. In the following section we propose a general technique for obtaining starting points for the solution of operator equations.

4. A PROCEDURE TO OBTAIN GOOD STARTING POINTS FOR NEWTON'S METHOD

In this section we consider $f: D \subset B_1 \to B_2$, where B_1 and B_2 are Banach spaces and assume that f is Fréchet differentiable. We shall give a procedure to obtain a point x_0 such that Newton's method starting from x_0 will converge to a root α of $f(x) = 0$, provided that f satisfies some natural conditions. The use of Newton's method is only illustrative. The principle of the procedure can be applied to other iterative methods.

Let $S_r(x_0)$ denote a ball in D with center x_0 and radius r. Sufficient conditions for the quadratic convergence of Newton's method, starting from x_0, are given by the famous Newton-Kantorovich Theorem (see, e.g., Ortega and Rheinboldt [70, Section 12.6.2]), which essentially states the following:

If

(4.1) $[f'(x_0)]^{-1}$ exists, $\|[f'(x_0)]^{-1}\| \le \beta_0$,

(4.2) $\|[f'(x_0)]^{-1} f(x_0)\| \le \xi_0$,

(4.3) $\|f'(x)-f'(y)\| \le K\|x-y\|$, $x,y \in S_r(x_0)$

and if

(4.4) $h_0 \equiv \beta_0 K \xi_0 < \frac{1}{2}$,

(4.5) $2\xi_0 \le r$

then Newton's method, starting from x_0, will generate iterates converging quadratically to a root α of $f(x) = 0$ and

(4.6) $\|x_0-\alpha\| \le 2\xi_0$.

For convenience, we say x_0 is a <u>good starting point</u> for approximating α by Newton's method or a good starting point for short, if conditions (4.1) ~ (4.5) are satsified. Note that the existence of a good starting point implies the existence of a zero of f in $S_q(x_0)$, $q = 2\xi_0$. The conditions for a point to be a good starting point are quite restrictive for certain applications. We are interested in designing algorithms for finding good starting points under relaxed conditions. We shall first prove the following existence theorem by combining the ideas of the Newton-Kantorovich Theorem and the continuation method (see, e.g. Ortega and Rheinboldt [70,

Section 7.5]). The algorithm (Algorithm 4.1) used in the proof is then developed to be a procedure (Algorithm 4.2) to obtain good starting points, for f satisfying some natural conditions.

Theorem 4.1. If f' satisfies a Lipschitz condition on $S_{2r}(x_0) \subseteq D$,

$$\|f(x_0)\| \leq \eta_0,$$

$$\|[f'(x)]^{-1}\| \leq \beta \text{ for all } x \in S_r(x_0), \text{ and}$$

$$\beta\eta_0 < r/2, \tag{4.7}$$

then there exists a root of $f(x) = 0$ in $S_r(x_0)$.

Proof. We assume that

$$\|f'(x)-f'(y)\| \leq K\|x-y\|, \quad x, y \in S_{2r}(x_0).$$

The proof is based on the following algorithm.

Algorithm 4.1.

The algorithm takes f satisfying the conditions of Theorem 4.1 as input and produces a good starting point for approximating a root α of $f(x) = 0$.

1. Set $h_0 \leftarrow \beta^2 K\eta_0$ and $i \leftarrow 0$.
   Pick any number δ in $(0,\frac{1}{2})$.

2. If $h_i < \frac{1}{2}$, x_i is a good starting point for approximating α and Algorithm 4.1 terminates.

3. Set $\lambda_i \leftarrow (\frac{1}{2}-\delta)/h_i$, and

$$f_i(x) \leftarrow [f(x)-f(x_i)] + \lambda_i f(x_i). \tag{4.8}$$

4. (It will be shown later that x_i is a good starting point for approximating a zero, denoted by x_{i+1}, of f_i.) Apply Newton's method to f_i, starting from x_i, to find x_{i+1}.

5. (Assume that the exact x_{i+1} is found.) Set $\eta_{i+1} \leftarrow \|f(x_{i+1})\|$ and $h_{i+1} \leftarrow \beta^2 K \eta_{i+1}$.

6. Set $i \leftarrow i+1$ and return to step 2.

In the following we prove the correctness of the algorithm. First we note that $\lambda_i \in (0,1)$ and by (4.8)

$$\eta_{i+1} = (1-\lambda_i)\eta_i. \tag{4.9}$$

We shall prove by induction that

$$\|x_i - x_{i-1}\| \le 2\beta\lambda_{i-1}\eta_{i-1}, \text{ and} \tag{4.10}$$

$$\|x_i - x_0\| \le r. \tag{4.11}$$

They trivially hold for $i = 0$.

Suppose that (4.10) and (4.11) hold and $h_i \ge \frac{1}{2}$. By (4.8),

$$\beta^2 K\|f_i(x_i)\| \le \beta^2 K\lambda_i\eta_i = \lambda_i h_i = \frac{1}{2}-\delta, \tag{4.12}$$

and by (4.9),

$$2\beta\|f_i(x_i)\| \le 2\beta\lambda_i\eta_i < 2\beta\eta_i \le 2\beta\eta_0 < r.$$

Further, by (4.11), we have $S_r(x_i) \subseteq S_{2r}(x_0)$. Hence x_i is a good starting point for approximating the zero x_{i+1} of f_i. From (4.6), we know

$$\|x_{i+1} - x_i\| \le 2\beta\lambda_i\eta_i . \tag{4.13}$$

Hence (4.10) holds with i replaced by i+1. By (4.13), (4.9) and (4.7), we have

$$(4.14)\quad \begin{aligned} \|x_{i+1}-x_0\| &\le \|x_{i+1}-x_i\| + \|x_i-x_{i-1}\| + \cdots + \|x_1-x_0\| \\ &\le 2\beta(\lambda_i\eta_i + \lambda_{i-1}\eta_{i-1} + \cdots + \lambda_0\eta_0) \\ &\le 2\beta((1-\lambda_{i-1})\eta_{i-1} + \lambda_{i-1}\eta_{i-1} + \cdots + \lambda_0\eta_0) \\ &= 2\beta(\eta_{i-1} + \lambda_{i-2}\eta_{i-2} + \cdots + \lambda_0\eta_0) \\ &\le \cdots \\ &\le 2\beta\eta_0 < r, \end{aligned}$$

i.e., (4.11) holds with i replaced by i+1.

We now assume that (4.10) and (4.11) hold and $h_i < \frac{1}{2}$. By (4.7) and (4.9), $2\beta\|f(x_i)\| = 2\beta\eta_i < 2\beta\eta_0 < r$. Further by (4.11), $S_r(x_i) \subseteq S_{2r}(x_0)$. Hence x_i is a good starting point for approximating α.

It remains to show that the loop starting from step 2 is finite. Suppose that $h_0 \ge \frac{1}{2}$. Since $\lambda_i \in (0,1)$ for all i, we have

$$\lambda_i = \frac{\frac{1}{2}-\delta}{\beta^2 K\eta_i} = \frac{\frac{1}{2}-\delta}{\beta^2 K(1-\lambda_{i-1})\eta_{i-1}} > \frac{\frac{1}{2}-\delta}{\beta^2 K\eta_{i-1}} = \lambda_{i-1}, \text{ for all i.}$$

Hence by (4.9) $\eta_{i+1} = (1-\lambda_i)\eta_i < (1-\lambda_0)\eta_i$

$$< \cdots$$

$$< (1-\lambda_0)^{i+1}\eta_0.$$

This implies that $h_i < \frac{1}{2}$ when $\beta^2 K(1-\lambda_0)^i\eta_0 < \frac{1}{2}$, i.e., when

$$(4.15)\quad (1-\lambda_0)^i < \frac{1}{2h_0}.$$

Since $1-\lambda_0 < 1$, (4.15) is satisfied for large i. Therefore when i is large enough, $h_i < \frac{1}{2}$ and hence Algorithm 4.1 terminates. The proof of Theorem 4.1 is complete. ■

Note that Theorem 4.1 is trivial for the scalar case (i.e., when $f: R \to R$), since the mean value theorem can be used there. The problem becomes nontrivial for nonscalar cases. The main reason for including Theorem 4.1 here is to introduce Algorithm 4.1, which works for Banach spaces. It should be noted that some assumptions (e.g., (4.7)) of the theorem could be weakened by complicating the algorithm used in the proof. Theorem 4.1 is similar to a result of Avila [70, Theorem 4.3], where, instead of the assumption (4.7) used here, a more complicated assumption involving β, k, η_0 was used. Also the idea of his algorithm is basically different from that of Algorithm 4.1.

An upper bound $N(h_0,\delta)$ on the number of times the loop starting from step 2 is executed in Algorithm 4.1 can be obtained from (4.15). Since $\lambda_0 = (\frac{1}{2}-\delta)/h_0$, (4.15) is equivalent to

$$(4.16)\quad (1 - \frac{\frac{1}{2}-\delta}{h_0})^i \le \frac{1}{2h_0},$$

from which $N(h_0,\delta)$ can be calculated. Asymptotically, we have

$$(4.17)\quad N(h_0,\delta) \sim \frac{2h_0 \ln h_0}{1-2\delta}, \text{ as } h_0 \to \infty.$$

From (4.16) and (4.17) it appears that we should use small δ in the algorithm. However, small δ tends to slow down the convergence of the Newton iterates in step 4 of the algorithm (see (4.18), (4.19)). The problem of how to choose a suitable δ will be further discussed after Algorithm 4.2.

In Algorithm 4.1, we assume that the exact zero x_{i+1} of f_i can be found by Newton's method. This is clearly not the case in practice. Fortunately, this problem can be solved by modifying Algorithm 4.1. The modified algorithm, Algorithm 4.2, appears in the proof of the following theorem. In the theorem and the rest of the paper, a Newton step means the computation of $x - [f'(x)]^{-1}f(x)$, given x, f and f'. Hence a Newton step involves one evaluation of f, one of f' at x and the computation of $x - [f'(x)]^{-1}f(x)$ from x, $f(x)$, $f'(x)$.

Theorem 4.2. Suppose that the conditions of Theorem 4.1 are satisfied and

$$\|f'(x)-f'(y)\| \le K\|x-y\|, \quad x,y \in S_{2r}(x_0).$$

Then a good starting point for approximating the root of $f(x) = 0$ by Newton's method can be obtained in $N(\delta)$ Newton steps, where δ is any number in $(0,\frac{1}{2})$ and $N(\delta)$ is defined as follows. If $h_0 = \beta^2 K\eta_0 \le \frac{1}{2}-\delta$ then $N(\delta) = 0$ else $N(\delta)=I(\delta)J(\delta)$, where $I(\delta)$ is the smallest integer i such that

$$(1-\frac{\frac{1}{2}-\delta}{h_0})^i \le (\frac{1}{2}-\delta)\frac{1}{h_0},$$

and $J(\delta)$ is the smallest integer j such that

$$(4.18)\quad \frac{1}{2^{j-1}}(1-2\delta)^{2^j-1}(m+\beta\eta_0) \le r - 2\beta\eta_0,$$

$$(4.19)\quad \frac{1}{2^j}(1-2\delta)^{2^j-1}(m+\beta\eta_0) \le m,$$

with $m = \min(\frac{r}{2}-\beta\eta_0, \frac{\delta}{2\beta K})$.

Proof. The proof is based on the following algorithm, which

is adapted from Algorithm 4.1.

Algorithm 4.2.

The algorithm takes f satisfying the conditions of Theorem 4.2 as input and produces a good starting point for approximating a root α of $f(x) = 0$.

1. Set $h_0 \leftarrow \beta^2 K\eta_0$, $\bar{x}_0 \leftarrow x_0$ and $i \leftarrow 0$.
   Pick any number δ in $(0,\frac{1}{2})$.

2. If $h_i \leq \frac{1}{2}-\delta$, $\bar{x}_i$ is a good starting point for approximating α and Algorithm 4.2 terminates.

3. Set $\lambda_i \leftarrow (\frac{1}{2}-\delta)/h_i$,

$$f_i(x) \leftarrow [f(x)-\eta_i f(x_0)/\eta_0] + \lambda_i \eta_i f(x_0)/\eta_0, \text{ and}$$

$$\eta_{i+1} \leftarrow (1-\lambda_i)\eta_i. \tag{4.20}$$

4. Apply Newton's method to f_i, starting from $\bar{x}_i$, to find an approximation $\bar{x}_{i+1}$ to a zero x_{i+1} of f_i such that

$$\|\bar{x}_{i+1}-x_{i+1}\| \leq r - 2\beta\eta_0, \text{ and} \tag{4.21}$$

$$\|[f_i'(\bar{x}_{i+1})]^{-1} f_i(\bar{x}_{i+1})\| \leq \min(\tfrac{r}{2}-\beta\eta_{i+1}, \tfrac{\delta}{2\beta K}) \tag{4.22}$$

5. Set $h_{i+1} \leftarrow \beta^2 K\eta_{i+1}$.

6. Set $i \leftarrow i+1$ and return to step 2.

Note that the h_i, λ_i, η_i, f_i, x_i in Algorithm 4.2 are the same h_i, λ_i, η_i, f_i, x_i in Algorithm 4.1. Note also that by (4.21) and (4.14) we have

$$\|\bar{x}_i-x_0\| \leq \|\bar{x}_i-x_i\| + \|x_i-x_0\| \tag{4.23}$$

$$\leq (r - 2\beta\eta_0) + 2\beta\eta_0 = r, \ \forall i.$$

It is clear that if $h_0 < \frac{1}{2} - \frac{\delta}{2}$, $\bar{x}_0$ is a good starting point for approximating α. Now suppose $h_0 > \frac{1}{2} - \frac{\delta}{2}$. Since $\bar{x}_0 = x_0$, in the proof of Theorem 4.1 we have shown that $\bar{x}_0$ is a good starting point for approximating x_1, a zero of f_0. Let z_j denote the jth Newton iterate starting from $\bar{x}_0$ for approximating x_1. Since

$$\beta^2 K \|f_0(\bar{x}_0)\| = \beta^2 K \lambda_1 \eta_0 = \frac{1}{2} - \delta,$$

it is known (see e.g. Rall [71, Section 22]) that

$$\|z_j - x_1\| \leq \frac{1}{2^{j-1}}(1-2\delta)^{2^j-1} \|[f'(x_0)]^{-1} f(x_0)\| \text{ and}$$

$$\|[f_0'(z_j)]^{-1} f_0(z_j)\| \leq \frac{1}{2^j}(1-2\delta)^{2^j-1} \|[f'(x_0)]^{-1} f(x_0)\|.$$

Hence we may let $\bar{x}_1$ be z_j for j large enough, say, $j=j(\delta)$, then

$$\|\bar{x}_1 - x_1\| \leq r - 2\beta\eta_0, \text{ and}$$

$$\|[f_0'(\bar{x}_1)]^{-1} f_0(\bar{x}_1)\| \leq \min(\tfrac{r}{2} - \beta\eta_1, \tfrac{\delta}{2\beta K}),$$

i.e., (4.21) and (4.22) hold for i = 0.

Suppose that (4.21) and (4.22) hold. Then

(4.24) $\|[f'(\bar{x}_{i+1})]^{-1} f(\bar{x}_{i+1})\|$

$\leq \|[f_i'(\bar{x}_{i+1})]^{-1} f_i(\bar{x}_{i+1})\| + \|[f_i'(\bar{x}_{i+1})]^{-1}[f(\bar{x}_{i+1}) - f_i(\bar{x}_{i+1})]\|.$

$\leq \min(\frac{r}{2} - \beta\eta_{i+1}, \frac{\delta}{2\beta K}) + \beta\eta_{i+1}$, and

(4.25) $\|[f_{i+1}'(\bar{x}_{i+1})]^{-1} f_{i+1}(\bar{x}_{i+1})\|$

$$\leq \|[f'(\bar{x}_{i+1})]^{-1}[f(\bar{x}_{i+1}) - \eta_{i+1}f(x_0)/\eta_0]\|$$

$$+ \|[f'(\bar{x}_{i+1})]^{-1}\lambda_{i+1}\eta_{i+1}f(x_0)/\eta_0]\|$$

$$\leq \|[f_i'(\bar{x}_{i+1})]^{-1}f_i(\bar{x}_{i+1})\| + \lambda_{i+1}\beta\eta_{i+1}.$$

Suppose that $h_{i+1} < \frac{1}{2} - \delta$. We want to prove that $\bar{x}_{i+1}$ is a good starting point for approximating α. By (4.24),

$$\beta K\|[f'(\bar{x}_{i+1})]^{-1}f(\bar{x}_{i+1})\|$$

$$\leq \beta K \cdot \frac{\delta}{2\beta K} + h_{i+1} < \frac{\delta}{2} + \frac{1}{2} - \delta = \frac{1}{2} - \frac{\delta}{2}.$$

Let $a = \|f'(\bar{x}_{i+1})]^{-1}f(\bar{x}_{i+1})\|$. If $x \in S_{2a}(\bar{x}_{i+1})$, then

$$\|x-x_0\| \leq \|x-\bar{x}_{i+1}\| + \|\bar{x}_{i+1}-x_0\| \tag{4.26}$$

$$\leq 2a + r$$

$$\leq 2(\frac{r}{2} - \beta\eta_{i+1} + \beta\eta_{i+1}) + r = 2r,$$

i.e., $x \in S_{2r}(x_0)$. Hence $\bar{x}_{i+1}$ is a good starting point for approximating α. We now assume that $h_{i+1} > \frac{1}{2} - \delta$, and want to prove that $\bar{x}_{i+1}$ is a good starting point for approximating x_{i+2}, a zero of f_{i+1}.

We have by (4.25) and (4.22),

$$\beta K\|[f_{i+1}'(\bar{x}_{i+1})]^{-1}f_{i+1}(\bar{x}_{i+1})\|$$

$$\leq \frac{\delta}{2} + \lambda_{i+1}\beta^2 K\eta_{i+1}$$

$$= \frac{\delta}{2} + \frac{1}{2} - \delta = \frac{1}{2} - \frac{\delta}{2}.$$

Let $b = \|[f'_{i+1}(\bar{x}_{i+1})]^{-1} f_{i+1}(\bar{x}_{i+1})\|$. If $x \in S_{2b}(\bar{x}_{i+1})$, as in (4.26) we can prove that $x \in S_{2r}(x_0)$. Hence $\bar{x}_{i+1}$ is a good starting point for approximating x_{i+2}. By the same argument as used for obtaining $\bar{x}_0$ and by (4.25), one can prove that if $\bar{x}_{i+2}$ is set to be the $J(\delta)$-th Newton iterate starting from $\bar{x}_{i+1}$, then

$$\|\bar{x}_{i+2} - x_{i+1}\| \leq r - 2\beta\eta_0, \text{ and}$$

$$\|[f'_{i+1}(\bar{x}_{i+2})]^{-1} f_{i+1}(\bar{x}_{i+2})\| \leq \min(\frac{r}{2} - \beta\eta_{i+2}, \frac{\delta}{2\beta K}),$$

i.e. (4.21) and (4.22) hold with i replaced by i+1. This shows that we need to perform at most $J(\delta)$ Newton steps at step 4 of Algorithm 4.2 to obtain each $\bar{x}_{i+1}$. Furthermore, from an inequality similar to (4.16) it is easy to see that the loop starting from step 2 of Algorithm 4.2 is executed at most $I(\delta)$ times. Therefore, for any $\delta \in (0,\frac{1}{2})$, to obtain a good starting point we need to perform at most $N(\delta)=I(\delta)\cdot J(\delta)$ Newton steps. The proof of Theorem 4.2 is complete. ■

We have shown that Algorithm 4.2 with parameter $\delta \in (0,\frac{1}{2})$ finds a starting point for Newton's method, with respect to f satisfying the conditions of Theorem 4.2, in $N(\delta)$ Newton steps. One should note that δ should not be chosen to minimize the complexity of Algorithm 4.2. Instead, δ should be chosen to minimize the complexity of the corresponding algorithm for finding root α of the equation $f(x) = 0$, which is defined as follows:

1. Search phase: Perform Algorithm 4.2.
2. Iteration phase: Perform Newton's method starting from the point obtained by Algorithm 4.2.

Note that the choice of δ determines the terminating condition of Algorithm 4.2 and hence determines when the search phase is switched to the iteration phase. Therefore the optimal time to switch can be obtained by choosing δ to minimize the sum of the complexities of the two phases.

An upper bound on the complexity of the search phase is the time needed for performing $N(\delta)$ Newton steps. Suppose that we went to approximate α to within ε, for a given $\varepsilon > 0$. It can be shown that an upper bound on the complexity of the iteration phase is the time needed for performing $T(\delta, \varepsilon)$ Newton steps, where $T(\delta,\varepsilon)$ is the smallest integer k such that

$$(4.27) \quad \frac{1}{2^{k-1}}(1-2\delta)^{2^k-1}(m+\beta\eta_0) \leq \varepsilon,$$

(see (4.18)). Therefore, we have proved the following result.

Theorem 4.3. If f satisfies the conditions of Theorem 4.2, then the time needed to locate a root of f(x) within a ball of radius ε is bounded above by the time needed to perform $R(\varepsilon)$ Newton steps, where $R(\varepsilon) = \min_{0<\delta\leq\frac{1}{2}}(N(\delta)+T(\delta,\varepsilon))$, $N(\delta)$ is defined in Theorem 4.2 and $T(\delta,\varepsilon)$ is defined by (4.27).

For large h_0, we know from (4.17) that $R(\varepsilon)$ grows like $O(h_0 \ln h_0)$ as $h_0 \to \infty$. For fixed h_0, we can calculate the values of $R(\varepsilon)$ numerically. We have computed the values of $R(\varepsilon)$ for f satisfying the conditions of Theorem 4.3 with

1. $\beta\eta_0 \leq \frac{4}{5}\cdot\frac{r}{2}$,
2. $1 \leq h_0 \equiv \beta^2 K\eta_0 \leq 10$,

and for ε equal to $10^{-i}r$, $1 \leq i \leq 10$. Table 1 reports the results for ε equal to $10^{-6}r$.

h_0	δ_0	$I(\delta_0)$	$J(\delta_0)$	$N(\delta_0)$	$T(\delta_0,\varepsilon)$	$R(\varepsilon)$
1	.165	3	2	6	5	11
2	.103	8	3	24	6	30
3	.118	16	3	48	6	54
4	.129	25	3	75	6	81
5	.137	35	3	105	6	111
6	.144	47	3	141	5	146
7	.149	59	3	177	5	182
8	.154	72	3	216	5	221
9	.159	85	3	255	5	260
10	.163	99	3	297	5	302

TABLE 1

In the table, δ_0 is the δ in $(0,\frac{1}{2})$ which minimizes $N(\delta) + T(\delta,\varepsilon)$, i.e., $R(\varepsilon) = T(\delta_0) + T(\delta_0,\varepsilon)$. Suppose, for example, that $h_0 = 9$ or $h_0 \leq 9$. Then by Algorithm 4.2 with $\delta = .159$ and by (4.27), we know that the search phase can be done in 255 Newton steps and the iteration phase in 5 Newton steps. Hence a root can be located within a ball of radius $10^{-6}r$ by 260 Newton steps.

5. SUMMARY AND CONCLUSIONS

The search and iteration phases should be studied together. A methodology for studying the worst case complexity of the two phases is proposed. Results based on the methodology are <u>global</u> and <u>non-asymptotic</u> (see Theorems 4.2 and 4.3), while the usual results in analytic computational complexity are local and asymptotic. Optimal time for switching from the search phase to the iteration phase can also be determined from the methodology. A useful lemma for proving lower bounds on complexity is identified. Several optimality results are

obtained for scalar functions.

The main results of the paper deal with the complexity of solving certain nonlinear operator equations $f(x) = 0$. Upper bounds are established by a new procedure for obtaining starting points for Newton's method. The procedure finds points where the conditions of the Newton-Kantorovich Theorem are satisfied. It is believed that the principle of the procedure can be used for other iterative schemes, where Newton-Kantorovich-like theorems are available, for f satisfying various kinds of conditions. It is not known, however, at this moment whether or not the number of Newton steps used by the procedure is close to the minimum. The problem of establishing lower bounds on the complexity for solving $f(x) = 0$ with f satisfying the conditions such as that of Theorem 4.3 deserves further research. The lemma mentioned above may be useful for the problem.

We end the paper by proposing an open question: Suppose that the conditions of the Newton-Kantorovich Theorem hold. Is Newton's method optimal or close to optimal, in terms of the numbers of function and derivative evolutions required to approximate the root to within a given tolerance?

ACKNOWLEDGMENTS

The author would like to thank J. F. Traub for his comments and G. Baudet, D. Heller, A. Werschulz for checking a draft of this paper.

REFERENCES

Avila [70] Avila, J., "Continuation Methods for Nonlinear Equations," Technical Report TR-142, Computer Science Center, University of Maryland, January, 1971.

Brent, Wolfe and Winograd [73] Brent, R. P., Winograd, S. and Wolfe, P., "Optimal Iterative Processes for Root-Finding, Numer. Math. 20 (1973), 327-341.

Micchelli and Miranker [75] Micchelli, C. A. and Miranker, W. L., "High Order Search Methods for Finding Roots," J.ACM 22 (1975), 51-60.

Ortega and Rheinboldt [70] Ortega, J. M. and Rheinboldt, W. C., Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

Rall [69] Rall, L. B., Computational Solution of Nonlinear Operator Equations, John Wiley and Sons, Inc., New York, 1969.

Traub and Woźniakowski [75] Traub, J. F. and Woźniakowski, H., "Strict Lower and Upper Bounds on Iterative Computational Complexity," to appear in Analytic Computational Complexity, edited by J. F. Traub, Academic Press.

Winograd [75] Winograd, S., "Optimal Approximation from Discrete Samples," to appear in Analytic Computational Complexity, edited by J. F. Traub, Academic Press.

Woźniakowski [74] Woźniakowski, H., "Maximal Stationary Iterative Methods for the Solution of Operator Equations," SIAM J. Numer. Anal. 11 (1974), 934-949.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE COMPLEXITY OF OBTAINING STARTING POINTS FOR SOLVING OPERATOR EQUATIONS BY NEWTON'S METHOD		5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) H. T. Kung		8. CONTRACT OR GRANT NUMBER(s) N00014-67-A-0314-0010, NR 044-422
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Department Pittsburgh, PA 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		12. REPORT DATE October 1975 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS (of this report) UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The complexity of searching for good starting points for iterations is studied. Global and non-asymptotic results are obtained: A useful lemma for proving lower bounds is identified, several optimal results are given for scalar equations, and upper bounds for operator equations are established by a new procedure for obtaining starting points for Newton's method.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)