COMPUTATIONAL COMPLEXITY OF
ONE-POINT AND MULTIPOINT ITERATION

H. T. Kung and J. F. Traub

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa.

April 1973

This research was supported in part by the National Science Foundation under Grant GJ32111 and the Office of Naval Research under Contract N00014-67-A-0314-0010, NR 044-422.

ABSTRACT

Let φ be an iteration for approximating the solution of a problem f. We define a new efficiency measure $e(\varphi,f)$. For a given problem f, we define the optimal efficiency $E(f)$ and establish lower and upper bounds for $E(f)$ with respect to different families of iterations. We conjecture an upper bound on $E(f)$ for any iteration without memory.

TABLE OF CONTENTS

1. INTRODUCTION

Let φ be an iteration for approximating the solution of a problem f. We define a new efficiency measure $e(\varphi,f)$. The efficiency measure gives us a methodology for comparing iterations as well as permitting us to derive theoretical limits on iteration efficiency.

For a given problem f, we define the optimal efficiency $E(f)$ over all φ belonging to a family Φ. We establish lower and upper bounds for $E(f)$ with respect to different families of iterations. We conjecture an upper bound on $E(f)$ for any iteration without memory.

We summarize the results of this paper. Basic concepts are given in Section 2 and our efficiency measure is defined in Section 3. In the next two sections we establish lower and upper bounds on the optimal efficiency for solving a problem with respect to important families of algorithms. A conjecture on optimal efficiency is stated in Section 6 and a small numerical example is given in the last section.

2. BASIC CONCEPTS

We work over the field of real numbers. Let $\sigma(x)$ be a function and λ_σ be a procedure which computes the value of $\sigma(x)$ for any given x. (We write λ for λ_σ if there is no ambiguity.) Let α be any number. We say $\Sigma = (\sigma,\lambda)$ is an <u>algorithm</u> for approximating α if the sequence $\{x_i\}$, generated by $x_{i+1} = \sigma(x_i)$, converges to α whenever x_0 is chosen near α; and if $\sigma(x_i)$ is computed by the procedure λ for all i. $\Sigma = (\sigma,\lambda)$ has <u>order of convergence</u> $p(\sigma)$ if

$$\lim_{x\to\alpha} \frac{\sigma(x)-\alpha}{(x-\alpha)^{p(\sigma)}}$$

exists and is non-zero. We measure the goodness of the algorithm $\Sigma = (\sigma,\lambda)$ by $p(\sigma)$ and define the efficiency of the algorithm $\Sigma = (\sigma,\lambda)$ to be

$$(2.1) \quad e(\Sigma) = \frac{\log p(\sigma)}{c(\lambda)},$$

where $c(\lambda)$ is the cost of performing the procedure λ. In this paper we consider only superlinear convergent algorithms, that is, $p(\sigma) > 1$. All logarithms are to base 2.

For any fixed positive integer n, consider the algorithm $\Sigma_n = (\sigma_n,\lambda_n)$ where $\sigma_n = \underbrace{\sigma\circ\sigma\circ\cdots\circ\sigma}_{n\text{ times}}$, ($\sigma\circ\sigma$ denotes composition) and λ_n is the procedure which computes $\sigma_n(x)$ by

$$y_0 = x,$$
$$y_{i+1} = \sigma(y_i),\ i=0,\ldots,n-1,$$
$$\sigma_n(x) = y_n,$$

with $\sigma(y_i)$ being computed by λ for all i. One can easily check that

$p(\sigma_n) = p^n(\sigma)$ and $c(\lambda_n) = nc(\lambda_n)$. Note that

$$\frac{\log p(\sigma)}{c(\lambda)} = \frac{\log p^n(\sigma)}{nc(\lambda)} .$$

Therefore,

$$e(\Sigma) = e(\Sigma_n)$$

for any n. This invariance is clearly desirable for any useful efficiency measure, since Σ_n is just the algorithm which repeats Σ n times and hence Σ and Σ_n must have the same efficiency. Gentleman [70] shows that if any efficiency measure satisfies this invariance property then it must be of the form (2.1) or a strictly increasing function of that form. Hence (2.1) is essentially the unique way to define an efficiency measure. Furthermore, Traub [64, Equation C-11] shows that if the efficiency measure has the form (2.1) then efficiency is inversely proportional to the total cost of approximating α by the algorithm. More specifically, let Σ^1, Σ^2 be two algorithms for approximating α and let $k(\Sigma^1)$, $k(\Sigma^2)$ be the total costs for generating two sequences which start with the same initial approximation and terminate when some fixed number of correct digits of α have been calculated. Then

$$(2.2)\quad \frac{k(\Sigma^1)}{k(\Sigma^2)} \sim \frac{e(\Sigma^2)}{e(\Sigma^1)} .$$

Therefore, it is desirable to have algorithms with high efficiency. <u>An algorithm is called optimal in a certain class of algorithms if it has the highest efficiency among all algorithms in that class.</u>

We now consider how to define the cost $c(\lambda)$. Paterson [72] defines $c(\lambda)$ as the number of multiplications or divisions, except by constants, needed to perform the procedure λ. We call the associated efficiency the multiplicative efficiency. Kung [72] shows that unity is the sharp upper bound on the multiplicative efficiency, and Kung [73] uses the multiplicative efficiency to investigate the computational complexity of algebraic numbers. In this paper, we define $c(\lambda)$ to be the number of arithmetic operations needed to perform the procedure λ.

3. EFFICIENCY MEASURE FOR ITERATION

In the previous section we have defined the efficiency of an algorithm for approximating a number α. More specifically, we now study the efficiency of an algorithm for approximating a simple zero α_f of a function $f \in D$, where D is the set of analytic functions f which have simple zeros α_f. We consider algorithms $\Sigma = (\sigma,\lambda)$ where $\sigma = \varphi(f)$, φ is a one-point or multipoint iteration and $f \in D$. (See Kung and Traub [73].) If φ is a k-point iteration, $k=1,2,\ldots$, then φ has the following property:

For $j=0,\ldots,k-1$ there exists a function $u_{j+1}(y_0;y_1^0,\ldots,y_{d_0+1}^0;\ldots;y_1^j,\ldots,y_{d_j+1}^j)$ of $1+\sum_{i=0}^{j}(d_i+1)$ variables such that for all $f \in D$, if

$$
(3.1)\quad \begin{cases} z_0(x) = x, \ x \text{ belongs to the domain of } \varphi(f), \\ y_{i+1}^j(x) = f^{(i)}(z_j(x)), \\ z_{j+1}(x) = u_{j+1}(x;\ y_1^0(x),\ldots,y_{d_0+1}^0(x);\ldots;y_1^j(x),\ldots,y_{d_j+1}^j(x)), \end{cases}
$$

for $j=0,\ldots,k-1$, $i=0,\ldots,d_j$, then

$$(3.2)\quad \varphi(f)(x) = z_k(x).$$

In this paper we assume that

(3.3) all u_j are rational functions;

(3.4) if f is transcendental, we use a rational subroutine to approximate $f^{(i)}$, $i \geq 0$, whenever $f^{(i)}$ is transcendental; and

(3.5) all $f^{(i)}(z_j(x))$ are algebraically independent.

Assumption (3.5) means that we are not allowed to use any special property of f. In other words, we consider "general" f.

Recall that $\lambda(=\lambda_{\varphi(f)})$ is a procedure which computes the value of $\varphi(f)(x)$ for any x. Because of (3.5), λ must compute $\varphi(f)(x)$ according to (3.1) and (3.2). Let $a_j(\lambda)$, $j=1,\dots,k$, denote the number of arithmetic operations needed to compute $u_j(y_0;y_1^0,\dots,y_{d_{j-1}+1}^{j-1})$ for given $(y_0;y_1^0,\dots,y_{d_{j-1}+1}^{j-1})$ by the procedure λ. Moreover, if $f^{(i)}$ is rational, let $c(f^{(i)})$ denote the number of arithmetic operations for one evaluation of $f^{(i)}$; otherwise let $c(f^{(i)})$ denote the number of arithmetic operations used in the rational subroutine which approximates $f^{(i)}$. Then the total number of arithmetic operations needed to perform the procedure λ is

$$c(\lambda) = \sum_{i\geq 0} v_i(\varphi)c(f^{(i)}) + \sum_{i=1}^{k} a_i(\lambda)$$

where $v_i(\varphi)$ is the number of evaluations of $f^{(i)}$ required by φ.

If $p(\varphi)$ is the order of convergence of the iteration φ, then by definition (2.1) the efficiency of the algorithm $(\varphi(f),\lambda)$ is

$$e(\varphi(f),\lambda) = \frac{\log p(\varphi)}{c(\lambda)} = \frac{\log p(\varphi)}{\sum_{i\geq 0} v_i(\varphi)c(f^{(i)}) + \sum_{i=1}^{k} a_i(\lambda)} .$$

We define $e(\varphi,f)$, the <u>efficiency of the iteration φ with respect to the problem f</u>, by

$$e(\varphi,f) = \sup_{\lambda} e(\varphi(f),\lambda).$$

Let

$$a(\varphi) = \min_{\lambda} \sum_{i=1}^{k} a_i(\lambda).$$

Then

$$(3.6)\quad \boxed{e(\varphi,f) = \frac{\log p(\varphi)}{\sum_{i\geq 0} v_i(\varphi)c(f^{(i)})+a(\varphi)}}.$$

This is the basic efficiency measure used in this paper.

Define

$$\sum_{i\geq 0} v_i(\varphi)c(f^{(i)})$$

to be the evaluation cost of φ with respect to f and define $a(\varphi)$ to be the combinatory cost of φ. The total cost, which appears in the denominator of (3.6), is the sum of these two costs.

Let

$$(3.7)\quad c_f = \min_{i\geq 0} c(f^{(i)}).$$

In this paper, we refer to c_f as the problem complexity. Let

$$(3.8)\quad v(\varphi) = \sum_{i\geq 0} v_i(\varphi).$$

Then by (3.6),

$$(3.9)\quad e(\varphi,f) \leq \frac{\log p(\varphi)}{v(\varphi)c_f+a(\varphi)}.$$

This gives an upper bound on $e(\varphi,f)$.

The efficiency measure defined by (3.6) is the first one to include both evaluation and combinatory costs. Ostrowski [66, Chapter 3] defines efficiency as $p(\varphi)^{\frac{1}{v(\varphi)}}$ where $v(\varphi)$ is defined by (3.8). This amounts to neglecting $a(\varphi)$ and taking $c(f^{(i)})$ to be unity for all i in (3.6). Our efficiency measure, defined by (3.6), does not take into account rounding errors or truncation errors caused by rational approximations for transcendental $f^{(i)}$, $i \geq 0$.

The following two examples illustrate the definitions.

Example 3.1. (Newton-Raphson Iteration)

$$\varphi(f)(x) = x - \frac{f(x)}{f'(x)} .$$

This is a one-point iteration with $p(\varphi) = 2$, $v_0(\varphi) = v_1(\varphi) = 1$, and $a(\varphi) = 2$. Hence

$$e(\varphi,f) = \frac{1}{c(f)+c(f')+2} ,$$

$$e(\varphi,f) \le \frac{1}{2c_f+2} .$$

Example 3.2.

$$z_0 = x,$$

$$z_1 = z_0 - \frac{f(z_0)}{f'(z_0)},$$

$$\varphi(f)(x) = z_1 - \frac{f(z_1)f(z_0)}{[f(z_1)-f(z_0)]^2} \cdot \frac{f(z_0)}{f'(z_0)}.$$

This is a two-point iteration with $p(\varphi) = 4$, $v_0(\varphi) = 2$, $v_1(\varphi) = 1$ and $a(\varphi) = 8$. (See Kung and Traub [73, Section 5].) Hence

$$e(\varphi,f) = \frac{2}{2c(f)+c(f')+8},$$

$$e(\varphi,f) \le \frac{2}{3c_f+8}.$$

It is natural to ask for a given problem f what is the optimal value of $e(\varphi,f)$ for all φ belonging to some family Φ. Define

$$E_n(\Phi,f) = \sup_{\varphi\in\Phi}\{e(\varphi,f) \mid v(\varphi) = n\}.$$

Thus $E_n(\Phi,f)$ is the optimal efficiency over all $\varphi \in \Phi$ which use n evaluations. Define

$$E(\Phi,f) = \sup\{E_n(\Phi,f) \mid n = 1,2,\ldots\}.$$

Thus $E(\Phi,f)$ is the optimal efficiency for all $\varphi \in \Phi$. We will establish lower and upper bounds for $E_n(\Phi,f)$ and $E(\Phi,f)$ with respect to different families of iterations. When there is no ambiguity, we write $E_n(\Phi,f)$ and $E(\Phi,f)$ as $E_n(f)$ and $E(f)$, respectively. Since in practice we are more concerned with efficiency for problems f with higher complexity, we are particularly interested in the asymptotic behavior of these bounds as $c_f \to \infty$.

4. THEOREMS ON EFFICIENCY OF ONE-POINT ITERATION

We consider first the family of one-point iteration $\{\gamma_n\}$. (See Kung and Traub [73, Section 3].) The important properties of $\{\gamma_n\}$ from our point of view are summarized in the following theorem proven by Traub [64, Section 5.1].

Theorem 4.1.

1. $v_i(\gamma_n) = 1$, $i=0,\ldots,n-1$, $v_i(\gamma_n) = 0$, $i > n-1$. Hence $v(\gamma_n) = n$.
2. $p(\gamma_n) = n$.

We now turn to an upper bound for $a(\gamma_n)$. Suppose that we have already obtained $f^{(i)}(x)$, $i=0,\ldots,n-1$ and we want to use them to form $\gamma_n(f)(x)$. This amounts to calculating the first n-1 derivatives of f^{-1} (the inverse function) at $f(x)$. This can be done in $O(n^3)$ arithmetic operations by the power series reversion technique reported in Knuth [1969, Section 4.7]. However if one uses the Fast Fourier Transform for polynomial multiplication then the power series reversion can be done in $O(n^2 \log n)$ arithmetic operations, and this implies that

$$(4.1)\quad a(\gamma_n) \le \rho\, n^2 \log n$$

for some positive constant ρ. Then by (4.1) and Theorem 4.1,

$$(4.2)\quad e(\gamma_n,f) \ge \frac{\log n}{\sum_{i=0}^{n-1} c(f^{(i)})+\rho n^2 \log n}.$$

For n small, $a(\gamma_n)$ can be calculated by inspection. For instance, since

$$\gamma_3(f)(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)}{2f'(x)}\left[\frac{f(x)}{f'(x)}\right]^2 ,$$

one can easily observe that $a(\gamma_3) = 7$. Hence

$$(4.3) \quad e(\gamma_3,f) = \frac{\log 3}{c(f)+c(f')+c(f'')+7} .$$

Let φ be any one-point iteration, with $v(\varphi) = n$, which satisfies a mild smoothness condition. Then by Traub [64, Section 5.4], Kung and Traub [73, Theorem 6.1] $v_i(\varphi) \geq 1$, $i=0,\ldots,p(\varphi)-1$, and hence $p(\varphi) \leq n$. Clearly, $a(\varphi) \geq n-1$. Therefore, from (3.9),

$$(4.4) \quad e(\varphi,f) \leq \frac{\log n}{nc_f+n-1} \equiv h(n).$$

It is straightforward to verify that

$$(4.5) \quad h(n) \leq \frac{\log 3}{3c_f+2} , \text{ for } c_f > 4.$$

From (4.2), (4.3), (4.4) and (4.5) we have

Theorem 4.2.

For the family Φ of one-point iterations,

$$(4.6) \quad \frac{\log n}{\sum_{i=0}^{n-1} c(f^{(i)})+\rho n^2\log n} \leq E_n(f) \leq \frac{\log n}{nc_f+n-1}, \text{ for a constant } \rho > 0, \forall n,$$

$$(4.7) \quad \frac{\log 3}{c(f)+c(f')+c(f'')+7} \leq E(f) \leq \frac{\log 3}{3c_f+2}, \text{ for } c_f > 4.$$

Remark 4.1.

1. In (4.6) both lower and upper bounds for $E_n(f)$ are tight for f such that $c(f^{(i)}) \sim c_f$, $i < n$, and c_f is large, since lower bound/upper

bound $\to 1$ as $c_f \to \infty$.

2. For f such that $c(f) \sim c(f') \sim c(f'') \sim c_f$, and c_f is large, both lower and upper bounds for $E(f)$ in (4.7) are tight, since lower bound/upper bound $\to 1$ as $c_f \to \infty$. In this case, by (4.3), γ_3 is close to optimal among all one-point iterations.

5. THEOREMS ON EFFICIENCY OF MULTIPOINT ITERATION

We consider first the family of iterations $\{\Psi_n\}$ defined by Kung and Traub [73, Section 4]. The important properties of $\{\Psi_n\}$ from our point of view are summarized in

Theorem 5.1.

1. $v_0(\Psi_n) = n.\quad v_i(\Psi_n) = 0,\ i > 0.$ Hence $v(\Psi_n) = n$.
2. $p(\Psi_n) = 2^{n-1}$.

Kung and Traub [73, Appendix I] give a procedure λ for computing $\Psi_n(f)(x)$. It can be shown that

$$\sum_{j=1}^{n} a_j(\lambda) = \frac{3}{2}n^2 + \frac{3}{2}n - 7.$$

Hence

$$a(\Psi_n) \le \frac{3}{2}n^2 + \frac{3}{2}n - 7.$$

More generally, we assume that

$$(5.1)\quad a(\Psi_n) \le r(n),$$

where $r(n) = r_2n^2 + r_1n + r_0,\ r_2 > 0$.

Then by (5.1) and Theorem 5.1,

$$(5.2)\quad e(\Psi_n,f) \ge \frac{n-1}{nc(f)+r(n)}.$$

We choose n so as to maximize the right hand side of (5.2). The maximum is achieved when n=t where

$$t = 1 + \sqrt{\frac{c(f)}{r_2} + \delta},\ \delta = \frac{r_0+r_1+r_2}{r_2}.$$

Let

(5.3) $M = \mathrm{round}(t)$.

Then from (5.2) we can easily prove

Theorem 5.2.

There exists a constant $\zeta < 0$ such that if $M = M(f)$ is chosen by (5.3) then

$$e(\Psi_M, f) \geq \frac{1}{c(f)}\left(1 + \frac{\zeta}{\sqrt{c(f)}}\right), \text{ for } c(f) \text{ large.}$$

From (5.2) and Theorem 5.2, we have

Corollary 5.1.

For the family Φ of one-point or multipoint iterations,

$E_n(f) \geq \frac{n-1}{nc(f)+r(n)}$, where $r(n) = r_2n^2 + r_1n + r_0$, $r_2 > 0$; and

$E(f) \geq \frac{1}{c(f)}\left[1 + \frac{\zeta}{\sqrt{c(f)}}\right]$, for a constant $\zeta < 0$, for $c(f)$ large.

We turn to the family of iterations $\{\omega_n\}$ defined in Kung and Traub [73, Section 5]. The important properties of $\{\omega_n\}$ from our point of view are summarized in

Theorem 5.3.

1. $v_0(\omega_n) = n-1$, $v_1(\omega_n) = 1$, $v_i(\omega_n) = 0$, $i > 1$. Hence $v(\omega_n) = n$.
2. $p(\omega_n) = 2^{n-1}$.

Kung and Traub [73, Appendix I] give a procedure λ for computing $\omega_n(f)(x)$. It can be shown that

$$\sum_{j=1}^{n} a_j(\lambda) = \frac{3}{2}n^2 + \frac{3}{2}n - 4.$$

Hence

$$a(\omega_n) \le \frac{3}{2} n^2 + \frac{3}{2} n - 4.$$

More generally, we assume that

(5.4) $a(\omega_n) \le s(n)$

where $s(n) = s_2 n^2 + s_1 n + s_0$, $s_2 > 0$. Then by (5.4) and Theorem 5.3,

(5.5) $e(\omega_n, f) \ge \frac{n-1}{(n-1)c(f)+c(f')+s(n)}$.

We choose n so as to maximize the right hand side of (5.5). Then the maximum is achieved when $n = u$, where

$$u = 1 + \sqrt{\frac{c(f')}{s_2} + \varepsilon}, \quad \varepsilon = \frac{s_0+s_1+s_2}{r_2}.$$

Let

(5.6) $N = \text{round}(u)$.

Then from (5.5) we can easily prove

Theorem 5.4.

There exists a constant $\eta > 0$ such that if $N = N(f)$ is chosen by (5.6) then

$$e(\omega_N, f) \ge \frac{1}{c(f)+\eta\sqrt{c(f')}}, \text{ for } c(f') \text{ large.}$$

From (5.5) and Theorem 5.4, we have

Corollary 5.2.

For the family Φ of one-point or multipoint iterations,

$E_n(f) \ge \frac{n-1}{(n-1)c(f)+c(f')+s(n)}$, where $s(n) = s_2 n^2 + s_1 n + s_0, s_2 > 0$; and

$E(f) \ge \frac{1}{c(f)+\eta\sqrt{(f')}}$, for a constant $\eta > 0$, for $c(f')$ large.

We turn to more general families of multipoint iterations. Let φ be a Hermite interpolatory iteration with $v(\varphi) = n$. Then $p(\varphi) \le 2^{n-1}$ (Kung and Traub [73, Corollary 7.1]). Clearly, $a(\varphi) \ge n-1$. Hence by (3.9),

$$(5.7) \quad e(\varphi,f) \le \frac{n-1}{nc_f+n-1} \le \frac{1}{c_f+1}.$$

Since Ψ_n and ω_n are Hermite interpolatory iterations, from (5.7) and Corollaries 5.1, 5.2, we have

Theorem 5.5.

For the family Φ of Hermite interpolatory iterations,

$$\max\left(\frac{n-1}{nc(f)+r(n)}, \frac{n-1}{(n-1)c(f)+c(f')+s(n)}\right) \le E_n(f) \le \frac{n-1}{nc_f+n-1}, \ \forall n,$$

$$\max\left(\frac{1}{c(f)}\left[1+\frac{\zeta}{\sqrt{c(f)}}\right], \frac{1}{c(f)+\eta\sqrt{c(f')}}\right) \le E(f) \le \frac{1}{c_f+1},$$

for c_f large, where $r(n) = r_2n^2 + r_1n + r_0$, $r_2 > 0$, $s(n) = s_2n^2 + s_1n + s_0$, $s_2 > 0$, $\zeta < 0$ and $\eta > 0$.

Remark 5.1.

The lower and upper bounds for $E_n(f)$ and $E(f)$ stated in Theorem 5.5 are tight for f such that $c(f) \sim c_f$ and c_f is large, since lower bound/upper bound $\to 1$ as $c_f \to \infty$. In this case, by Theorem 5.2, Ψ_M is close to optimal among all Hermite interpolatory iterations.

Now, let φ be any multipoint iteration which uses evaluations of f only. Let $v(\varphi) = n$. Then $p(\varphi) \le 2^n$ (Kung and Traub [73, Theorem 7.2]). Clearly, $a(\varphi) \ge n-1$. Hence

$$(5.8)\quad e(\varphi,f) \le \frac{n}{nc(f)+n-1} \le \frac{1}{c(f)}.$$

Since Ψ_n is a multipoint iteration which uses evaluations of f only, from (5.8) and Corollary 5.1, we have

Theorem 5.6.

<u>For the family Φ of multipoint iterations using values of f only,</u>

$$\underline{\frac{n-1}{nc(f)+r(n)} \le E_n(f) \le \frac{n}{nc(f)+n-1},\ \forall n,}$$

$$\underline{\frac{1}{c(f)}\left[1+\frac{\zeta}{\sqrt{c(f)}}\right] \le E(f) \le \frac{1}{c(f)},}$$

<u>for c(f) large, where $r(n) = r_2 n^2 + r_1 n + r_0$, $r_2 > 0$, and $\zeta < 0$.</u>

Remark 5.2.

The lower and upper bounds for $E_n(f)$ and $E(f)$ stated in Theorem 5.6 are tight for f such that c(f) is large, since lower bound/upper bound → 1 as $c(f) \to \infty$. In this case, by Theorem 5.2, Ψ_M is close to optimal among all multipoint iterations using values of f only.

Remark 5.3.

For a given problem f let E'(f), E''(f) be the optimal efficiency achievable by one-point iteration and multipoint iteration, respectively. By Theorem 4.2 and Corollary 5.1,

$$E'(f) \le \frac{\log 3}{3c_f+2},$$

$$E''(f) \ge \frac{1}{c(f)}\left[1+\frac{\zeta}{\sqrt{c(f)}}\right],\ \zeta < 0,\ \text{for } c(f) \text{ large}.$$

Hence

$$\frac{E''(f)}{E'(f)} \geq \frac{3c_f+2}{(\log 3)c(f)}\left[1 + \frac{\zeta}{\sqrt{c(f)}}\right] \sim \frac{3}{\log 3} \cdot \frac{c_f}{c(f)}, \text{ for } c(f) \text{ large.}$$

In particular, if f is a problem such that $c_f = c(f)$ and c_f is large, then the ratio between optimal efficiencies achievable by multipoint iteration and one-point iteration is at least $\frac{3}{\log 3} \sim 1.89$.

6. A CONJECTURE

Kung and Traub [73] conjecture that if φ is any multipoint iteration with $v(\varphi) = n$ then $p(\varphi) \le 2^{n-1}$. Suppose that this conjecture is true. Then by (3.9), for any multipoint iteration φ with $v(\varphi) = n$,

$$e(\varphi,f) \le \frac{n-1}{nc_f+a(\varphi)} .$$

Clearly, $a(\varphi) \ge n-1$. Hence

$$e(\varphi,f) \le \frac{n-1}{nc_f+n-1} \equiv k(n).$$

Observe that

$$k(n) \le \frac{1}{c_f+1}, \forall n, \forall c_f.$$

Therefore we propose the following conjecture. It states, essentially, that the optimal efficiency for solving the problem f with respect to all one-point or multipoint iterations is bounded by the reciprocal of the problem complexity.

Conjecture 6.1.

For the family Φ of one-point or multipoint iterations,

$$\underline{E_n(f) \le \frac{n-1}{nc_f+n-1}} ,$$

$$\underline{E(f) \le \frac{1}{c_f+1}}.$$

7. NUMERICAL EXAMPLE

Let $f(x) = \sum_{i=1}^{50} ix^i - 25$. We calculate its simple zero $\alpha = -1$. Calculations were done in double precision arithmetic on a DEC PDP-10 computer. About 16 digits are available in double precision. Numerical results show the following: Starting with $x_0 = -1.01$, to bring the error to about 10^{-16}, five Newton-Raphson iterations are required while one ω_6 iteration is required. (See Table 7.1.) We assume that we do not take advantage of the algebraic dependence of f and f' (see the assumption of (3.5)) and that we use Horner's rule for the evaluation of f and f', treating each as an independent polynomial. Suppose that we use the procedure given by Kung and Traub [73, Appendix I] to compute $\omega_6(f)(x)$.

Let Σ^1 and Σ^2 be algorithms associated to Newton-Raphson iteration and ω_6 respectively. Then the total costs are

$$k(\Sigma^1) = 5[2\cdot 50 + 2\cdot 49 + 2] = 10^3,$$
$$k(\Sigma^2) = 5\cdot 2\cdot 50 + 2\cdot 49 + \frac{3}{2}\cdot 6^2 + \frac{3}{2}\cdot 6 - 4 = 657;$$

and the efficiencies are

$$e(\Sigma^1) = 1/[2\cdot 50 + 2\cdot 49 + 2] = 5/10^3,$$
$$e(\Sigma^2) = 5/[5\cdot 2\cdot 50 + 2\cdot 49 + \frac{3}{2}\cdot 6^2 + \frac{3}{2}\cdot 6 - 4 = 5/657.$$

Then

$$\frac{k(\Sigma^1)}{k(\Sigma^2)} = \frac{10^3}{657},$$

$$\frac{e(\Sigma^2)}{e(\Sigma^1)} = \frac{10^3}{657},$$

as predicted by (2.2). (In general, <u>approximate</u> equality holds from (2.2).)

Let $x_{i+1} = \varphi(x_i)$. The errors when φ is Newton-Raphson and $\varphi = \omega_6$ are shown in Table 7.1.

	Newton-Raphson	ω_6
$x_0-\alpha$	-1.0×10^{-2}	-1.0×10^{-2}
$x_1-\alpha$	-2.1×10^{-3}	-2.2×10^{-16}
$x_2-\alpha$	-1.0×10^{-4}	
$x_3-\alpha$	-2.7×10^{-7}	
$x_4-\alpha$	-1.8×10^{-12}	
$x_5-\alpha$	-1.1×10^{-16}	

Table 7.1

ACKNOWLEDGEMENT

We want to thank G. W. Stewart for his comments on the work reported in this paper.

BIBLIOGRAPHY

Gentleman [70] Gentleman, W. M., Private Communication, 1970.

Knuth [69] Knuth, D., The Art of Computer Programming, vol. 2, Seminumerical Algorithms, Addison-Wesley, Reading, Mass., 1969.

Kung [72] Kung, H. T., A Bound on the Multiplication Efficiency of Iteration, Proceedings of the Fourth Annual ACM Symposium on Theory of Computing. To appear in Journal of Computer and System Sciences, 1973.

Kung [73] Kung, H. T., The Computational Complexity of Algebraic Numbers, Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, 1973.

Kung and Traub [73] Kung, H. T. and Traub, J. F., Optimal Order of One-Point and Multipoint Iteration, report, Department of Computer Science, Carnegie-Mellon University, 1973.

Ostrowski [66] Ostrowski, A. M., Solution of Equations and Systems of Equations, 2nd ed., Academic Press, New York, 1966.

Paterson [72] Paterson, M. S., Efficient Iterations for Algebraic Numbers, in Complexity of Computer Computations, R. Miller and J. W. Thatcher (eds.), Plenum Press, New York, 1972, 44-52.

Traub [64] Traub, J. F., Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

Traub [72] Traub, J. F., Computational Complexity of Iterative Processes. SIAM Journal on Computing, vol. 1, no. 2, (1972), 167-179.