STRICT LOWER AND UPPER BOUNDS ON ITERATIVE COMPUTATIONAL COMPLEXITY*

J. F. Traub
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

H. Woźniakowski
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213
(Visiting from the University of Warsaw)

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

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

1. INTRODUCTION

Complexity is a measure of cost. The relevant costs depend on the model under analysis. The costs may be taken as units of time (in parallel computation), number of comparisons (in sorting algorithms), size of storage (in large linear systems), or number of arithmetics (in matrix multiplication). Of course a number of different costs may be relevant to a model. One can analyze the complexity of an algorithm, of a class of algorithms, or of a problem. The subject dealing with the complexity of an algorithm is usually called "Analysis of Algorithms". The subject dealing with the analysis of a class of algorithms or of a problem is called computational complexity.

Computational complexity comes in many flavors depending on the class of algorithms, the problem, and the costs. We limit ourselves here to mentioning three types of computational complexity. In each of these the costs are taken as the arithmetic operations. Algebraic computational complexity deals with a problem and a class of algorithms which solve the problems at finite cost. Typically the problem belongs to a class of problems which is indexed by an integer n. Let

$\text{comp}(P_n)$ be the complexity of solving the nth problem in the class. We are interested in lower bounds $L(P_n)$ and upper bounds $U(P_n)$ on $\text{comp}(P_n)$,

$$(1.1) \quad L(P_n) \le \text{comp}(P_n) \le U(P_n).$$

The upper bounds are obtained by exhibiting an algorithm for solving P_n with complexity $U(P_n)$. Lower bounds are obtained by theoretical considerations and "non-trivial" lower bounds are difficult to obtain. For example if P_n is the problem of multiplying two n by n matrices and if the cost of each arithmetic operation is taken as unity then

$$O(n^2) \le \text{comp}(P_n) \le O(n^\beta), \ \beta = \lg 7.$$

(We use lg to represent $\log_2$.) Borodin and Munro [75] survey the state of knowledge in algebraic complexity.

Exact solutions of "most" problems in science, engineering, and applied mathematics cannot be obtained with finite cost even if infinite-precision arithmetic is assumed. Indeed linear problems and evaluation of rational functions which can be solved at finite cost are the exception. Even when the problem can be solved rationally, we may choose to solve it by iteration. An example is the solution of large sparse linear systems. Typically, non-linear problems cannot be solved at finite cost.

We call the branch of complexity theory that deals with non-finite cost problems analytic computational complexity. Often the algorithms are iterative and we then refer to iterative computational complexity.

In this paper we propose a new methodology for iterative computational complexity. Our aim is to create at least a partial synthesis between iterative complexity and other types of complexity.

A basic quantity in iterative complexity has been the efficiency index of an algorithm or class of algorithms. In this paper we introduce a new quantity, the complexity index, which is the reciprocal of the efficiency index. The complexity index is directly proportional to the complexity of an algorithm or class of algorithms. We show under what conditions the complexity index is a good measure of complexity. Our methodology is non-asymptotic in the number of iterations. Earlier analyses of complexity applied only as the number of iterations went to infinity and this is not of course realistic in practice.

We summarize the contents of this paper. In Section 2 we analyze a simplified model of the errors of an iterative process and show that complexity is the product of two factors, the complexity index and the error coefficient function. Bounds on the error coefficient function are derived in the following Section and used to derive rigorous conditions for comparing the complexity of two different algorithms. In Section 4 we show how the results of the simple model can be applied to a realistic model of one-point iteration. Lower and upper bounds on the complexity index for several important classes of iterations appear in Section 5. In a short concluding Section we state the extensions and generalizations to be reported in future papers.

2. BASIC CONCEPTS

We analyze algorithms for the following problem. Let f be a non-linear real or complex scalar function with a simple zero α. Let x_0 be given and let an algorithm ϕ generate a sequence of approximations $x_1,\ldots,x_k$ to α. We terminate the algorithm when x_k is a sufficiently good approximation to α. This will be made precise below.

The appropriate setting for this investigation is to consider f as a non-linear operator on a Banach space of finite or infinite dimension. Since many of the basic ideas can be illustrated when f is a non-linear scalar function we shall assume throughout this paper that this holds. We must remark however that some of the most interesting and important results deal with the dependence of complexity on problem dimension and we do not deal with that dependence here.

Let $e_i > 0$ represent some measure of the error of x_i. For example, e_i might represent

$|x_i-\alpha|$, the absolute error

$\frac{|x_i-\alpha|}{|\alpha|}$, the relative error

$|f(x_i)|$, the residual.

Assume that the e_i satisfy the error equation

$$(2.1) \quad e_i = A_i e_{i-1}^p, \; p \geq 1, \; i = 1,2,\ldots,k.$$

We call p the <u>non-asymptotic order</u> and A_i the <u>error coefficient</u>. We require $0 < L \leq A_i \leq U < \infty$ for all values of e_0 including the possibility that e_0 be arbitrarily small. Then p is unique. Many iterations satisfy the model given by (2.1). In Section 6 we mention extensions to this model.

EXAMPLE 2.1. Let the algorithm be Newton-Raphson iteration and let e_i denote the absolute error. Then

$$p = 2, \; A_i = \left| \frac{f''(\eta_i)}{2f'(x_i)} \right|,$$

where η_i is in the interval spanned by α and x_i. ■

We simplify the model of (2.1) and show what kind of results may then be obtained. In Section 4 we return to the analysis of (2.1). Let

$$(2.2) \quad e_i = Ae_{i-1}^p, \; p \geq 1, \; i = 1,\ldots,k.$$

We call this the *constant error coefficient model* while (2.1) is the *variable error coefficient model*.

We consider first the case $p > 1$. It is easy to verify that

$$(2.3) \quad e_i = e_0\left(\frac{1}{w_p}\right)^{p^i - 1}, \; i = 0,\ldots,k,$$

where

$$(2.4) \quad w_p = \frac{1}{A^{\frac{1}{p-1}} e_0}.$$

Choose ε', $0 < \varepsilon' < 1$, and let k be the smallest index for which $e_k \leq \varepsilon' e_0$. Define $\varepsilon \leq \varepsilon'$ so that

$$(2.5) \quad e_k = \varepsilon e_0.$$

ε' is a basic parameter which measures the increase in precision to be obtained in the iteration. We choose ε to avoid ceiling and floor functions later in this paper. It is convenient to assume $\varepsilon \leq 2^{-2}$ (we use this in Theorem 3.1) but this is non-restrictive in practice.

From (2.3), (2.5),

(2.6) $$\varepsilon = \left(\frac{1}{w_p}\right)^{p^k-1},$$

and it follows that

(2.7) $$k = \frac{g(w_p)}{\lg p},$$

where

(2.8) $$g(w_p) = \lg\left(1 + \frac{t}{\lg w_p}\right),\quad t = \lg\left(\frac{1}{\varepsilon}\right).$$

This is independent of the logarithm base but it is convenient to take all logarithms to base 2. Then, if e_i is the relative error, t measures the number of bits to be gained in the iteration.

We denote the complexity of iteration i by c_i. In this paper we assume $c_i = c$ is independent of i. We defer a discussion of the estimation of c until Section 5. The important case of variable cost will be considered in a future paper. We define the complexity of the algorithm by

(2.9) $$\text{comp} = ck.$$

Then from (2.7), (2.8),

(2.10) $$\text{comp} = zg(w_p)$$

where we define

(2.11) $$z = \frac{c}{\lg p}.$$

as the complexity index.

We call g the error coefficient function. Equation (2.10) will be fundamental in our further analysis.

We have decomposed complexity into the product of two factors. The complexity index, which is independent of both the error coefficient and the starting error, is relatively easy to compute for any given algorithm. (However, lower bounds on the complexity index for classes of algorithms require upper bounds on order which is a difficult problem only solved for special cases (Kung and Traub [73], Meersman [75] and Woźniakowski [75b]).) We shall show, in a sense to be made precise in the next section, that the error coefficient is insensitive for a large portion of its domain and that complexity is determined primarily by the complexity index. We shall also show there are cases where complexity is determined primarily by the error coefficient function.

The complexity index is the reciprocal of a quantity called the efficiency index which has played an important role in iterative complexity. See, for example, Traub [64, Appendix C], Traub [72], Paterson [72] and Kung [73a]. Since complexity varies directly with the complexity index we feel that the complexity index rather than the efficiency index, should be basic.

We have been considering the case $p > 1$. For completeness we write down the case $p = 1$. Then $e_i = Ae_{i-1}$, $i = 1, 2,\ldots,k$ and $e_k = A^k e_0 = \varepsilon e_0$. Hence

$$(2.12)\quad k = \frac{t}{\lg(\frac{1}{A})}, \text{ comp } = \frac{ct}{\lg(\frac{1}{A})} .$$

We shall not pursue the case $p = 1$ further and shall assume for the remainder of this paper that $p > 1$, unless we state otherwise.

3. BOUNDS ON THE ERROR COEFFICIENT FUNCTION

We turn to an analysis of the error coefficient function which is one of the two factors which determines the complexity in (2.10). To see which values of w_p are of interest, note that from (2.3), $e_k < e_0$ iff $w_p > 1$. From the definition of k it is clear that $k \geq 1$ and hence from (2.7), (2.8), $w_p \leq \left(\frac{1}{\varepsilon}\right)^{\frac{1}{p-1}}$. Hence we assume

$$(3.1)\quad 1 < w_p \leq \left(\frac{1}{\varepsilon}\right)^{\frac{1}{p-1}} = 2^{\frac{t}{p-1}}.$$

Generally w_p depends on p. For many classes of iterations

$$(3.2)\quad a^{p-1} \leq A \leq b^{p-1}.$$

Then

$$\frac{1}{ae_0} \geq w_p \geq \frac{1}{be_0}$$

and the bounds on w_p are independent of p. If (3.2) holds for a class of iterations Φ we shall say that Φ is <u>normal</u>. An example of a normal class of iterations may be found in Woźniakowski [75b]. To simplify notation we shall henceforth write w_p as w whether or not we are dealing with a normal class.

Now, g(w) is a monotonically decreasing function and

$$\lim_{w\to 1^+} g(w) = \infty,\ \lim_{w\to\infty} g(w) = 0.$$

To study the size of g(w) we somewhat arbitrarily divide the range of w, given by (3.1), into three sub-ranges.

<u>$1 < w \leq 2$</u>. Since $g(w) = \lg(t + \lg w) - \lg\lg w$ and $0 < \lg w \leq 1$, we conclude

$$\lg t - \lg\lg w < g(w) \leq \lg(1+t) - \lg\lg w.$$

$\underline{2 \le w \le t}$. Since $g(w)\downarrow$, $g(2) = \lg(1+t)$, $g(t) > \lg t - \lg\lg t$, we conclude

$$\lg t - \lg\lg t < g(w) \le \lg(1+t).$$

$\underline{t \le w \le 2^{\frac{t}{p-1}},\ 2^{\frac{t}{p-1}} \ge t}$. Then

$$\lg p \le g(w) < 1 + \lg t - \lg\lg t$$

To get some feel for the length of these sub-ranges, observe that if e_i represents relative error then in single-precision computation on a "typical" digital computer we might take $\epsilon = 2^{-32}$. Then $t = 32$ and if $p = 2$, then $2^{\frac{t}{p-1}} = 2^{32}$.

From the bounds on the error coefficient function and (2.10) we immediately obtain the following bounds on complexity.

THEOREM 3.1. If $1 < w \le 2$,

$$z(\lg t - \lg\lg w) < \text{comp} \le z(\lg(1+t) - \lg\lg w). \tag{3.3}$$

If $2 \le w \le t$,

$$z(\lg t - \lg\lg t) \le \text{comp} \le z\lg(1+t). \tag{3.4}$$

If $t \le w \le 2^{\frac{t}{p-1}}$, (with $2^{\frac{t}{p-1}} \ge t$),

$$c \le \text{comp} < z(1 + \lg t - \lg\lg t). \tag{3.5}$$ ■

We discuss some of the implications of this Theorem. As w approaches unity, then for ϵ fixed, $\text{comp} \sim -z\lg\lg w$. In this case the effect of the error coefficient A and the initial error e_0 cannot be neglected.

Complexity depends more on the nearness of w to unity than of ϵ to zero. To see this, observe that if $2 \le w \le t$, $\text{comp} \sim z\lg\lg (1/\epsilon) = \text{comp}_1$ while if $1 < w \le 2$, $\text{comp} \sim z(\lg\lg (1/\epsilon) - \lg\lg w) = \text{comp}_2$. Let $\epsilon = 2^{-2^j}, w-1=2^{-2^j}\ln 2$. Then $\text{comp}_1 = jz$, $\text{comp}_2 \sim z(j+2^j)$.

Note that for any $p > 1$ the complexity of an iteration can be greater than if $p = 1$ (see (2.12)) provided w is sufficiently close to unity.

For any $w \ge 2$, complexity is bounded from above by $z\lg(1+t)$ and is therefore independent of the error coefficient A and the initial error e_0. For $w \ge 2$, complexity is insensitive to w and we need only crude bounds on w.

For $2 \le w \le t$,

$$1-\frac{\lg\lg t}{\lg t} \le \frac{\text{comp}}{z\lg t} \le 1+\frac{\lg(1+t^{-1})}{\lg t}$$

Therefore

$$1+o(1) \le \frac{\text{comp}}{z\lg t} \le 1+o(1)$$

and we conclude that on the interval $[2,t]$ we have, for t large, very tight bounds on comp with

$$(3.6) \quad \text{comp} \sim z\lg\lg \frac{1}{\epsilon} .$$

This should be compared with the case $p = 1$ (see (2.12)) where comp varies as $\lg \frac{1}{\epsilon}$.

We have taken $w = 2$ as one of our endpoints for convenience but this is of course arbitrary. Any value of w sufficiently far from unity will do. If $w = 2^{\frac{1}{\nu}}$ then $g(w) = \lg(1+\nu t)$. Then the effect of the nearness of w to unity and of ϵ to zero are equal if $\nu = t$, that is if $w = 2^{\frac{1}{t}}$. For this choice of w, $\text{comp} = z\lg(1+t^2) \sim 2z\lg t = 2z\lg\lg \frac{1}{\epsilon}$.

We have chosen the sub-ranges of w so that the endpoints are simple. We could also choose values of w that make the complexity formula simple. If

$$w = 2^{t/(t^{u}-1)},\ u \geq 1,\ \text{then comp} = uz\lg\lg \frac{1}{\epsilon},$$

while if

$$w = 2^{t/(t^{\frac{1}{\nu}}-1)},\ \nu \geq 1,\ \text{then comp} = \frac{1}{\nu}z\lg\lg \frac{1}{\epsilon}.$$

We now consider the methodology for comparing two iterations which are governed by the constant error coefficient model (2.2) and decrease the final error by the same ϵ. Let w_i, z_i, comp_i, $i = 1,2$ denote the parameters of the two iterations. Then

$$\frac{\text{comp}_1}{\text{comp}_2} = \left(\frac{z_1}{z_2}\right)\frac{g(w_1)}{g(w_2)}\ .$$

Clearly if $z_1 \leq z_2$ and $w_1 \geq w_2$ then $\text{comp}_1 \leq \text{comp}_2$. We obtain bounds on $\text{comp}_1/\text{comp}_2$ for sub-ranges of the w_i. Using the bounds on complexity from the previous theorem we obtain

THEOREM 3.2. If $1 < w_1, w_2 \leq 2$, then

$$(3.7)\quad \left(\frac{z_1}{z_2}\right)\left(\frac{\lg t-\lg\lg w_1}{\lg(1+t)-\lg\lg w_2}\right) < \frac{\text{comp}_1}{\text{comp}_2} < \left(\frac{z_1}{z_2}\right)\left(\frac{\lg(1+t)-\lg\lg w_1}{\lg t-\lg\lg w_2}\right).$$

If $1 < w_2 \leq 2 \leq w_1 \leq t$, then

$$(3.8)\quad \left(\frac{z_1}{z_2}\right)\left(\frac{\lg t-\lg\lg t}{\lg(1+t)-\lg\lg w_2}\right) \leq \frac{\text{comp}_1}{\text{comp}_2} < \left(\frac{z_1}{z_2}\right)\left(\frac{\lg(1+t)}{\lg t-\lg\lg w_2}\right).$$

If $2 \le w_1, w_2 \le t$, then

$$(3.9) \quad \left(\frac{z_1}{z_2}\right)\left(\frac{\lg t - \lg\lg t}{\lg(1+t)}\right) < \frac{\text{comp}_1}{\text{comp}_2} < \left(\frac{z_1}{z_2}\right)\left(\frac{\lg(1+t)}{\lg t - \lg\lg t}\right). \quad ■$$

We discuss some of the implications of this theorem. As $t \to \infty$, $\text{comp}_1/\text{comp}_2 \to z_1/z_2$ for any fixed values of w_1, w_2. The ratio z_1/z_2 has been the way that iterations have been compared (see Traub [64, Appendix C] where efficiency indices are used). Theorem 3.2 shows that z_1/z_2 can be a very poor measure of $\text{comp}_1/\text{comp}_2$; see for example (3.7).

Finally we observe that inequalities (3.7)-(3.9) can be rewritten to show when $\text{comp}_1 < \text{comp}_2$ or $\text{comp}_2 < \text{comp}_1$. For example, if $2 \le w_1, w_2 \le t$,

$$(3.10) \quad z_1 \le z_2\left(\frac{\lg t - \lg\lg t}{\lg(1+t)}\right), \text{ then } \text{comp}_1 < \text{comp}_2.$$

4. THE VARIABLE ERROR COEFFICIENT MODEL

We turn to the variable error coefficient model,

$$(4.1) \quad e_{i+1} = A_i e_i^p.$$

A complete analysis of this model is beyond the scope of this paper. Here we confine ourselves to the very simple assumption

$$(4.2) \quad A_L \le A_i \le A_U, \; i = 1,\dots,k.$$

Let

$$w_L = \frac{1}{\frac{1}{A_L^{p-1} e_0}}, \quad w_U = \frac{1}{\frac{1}{A_U^{p-1} e_0}}.$$

Then

(4.3) $zg(w_L) \leq \text{comp} \leq zg(w_U)$

Note that $w_U \leq w_L$ and therefore (4.3) is compatible with g being a monotonically decreasing function. We can now draw conclusions from the constant coefficient model with A replaced by A_L or A_U.

EXAMPLE 4.1. Let α be a real zero and let J denote an interval centered at α. Assume f' does not vanish in J and let $x_0 \in J$ and such that

$$e_0 = |x_0-\alpha| \leq \frac{\min\limits_{x\in J}|f'(x)|}{\max\limits_{x\in J}|f''(x)|} = \frac{1}{2A_U}$$

Then by Example 2.1, for Newton-Raphson iteration, $w_U \geq 2$ and <u>a priori</u>

(4.4) $\text{comp} \leq c \lg(1+t)$

The value of c is discussed in Section 5. Note that a sufficient condition for convergence is

$$e_0 < \frac{1}{A_U}$$

but with only this condition, complexity could be extremely large. ■

EXAMPLE 4.2. We seek to calculate $a^{1/2}$, that is solve $f(x) = x^2-a$. Let $a = 2^m\lambda^2$, m even, $\frac{1}{2} \leq \lambda^2 < 2$. Then $a^{1/2} = 2^{m/2}\lambda$, $1/2^{1/2} \leq \lambda < 2^{1/2}$. We use Newton-Raphson iteration,

$$x_{i+1} = \frac{1}{2}\left(x_i + \frac{\lambda^2}{x_i}\right).$$

Then $A_i = 1/(2x_{i-1})$. If $x_0 > \lambda$, then

$$A_L = \frac{1}{2x_0} \le A_i < \frac{1}{2\lambda} = A_U, \; i = 1,\ldots,k.$$

Hence

$$w_L = \frac{2}{1-\lambda/x_0}, \; w_U = \frac{2\lambda/x_0}{1-\lambda/x_0}.$$

Let $x_0 = 2^{1/2}$. Then $w_U \ge 2$ and $\text{comp} \le c \lg(1+t)$. To derive a lower bound on complexity one must make an assumption about the closest machine-representable number to $2^{1/2}$. We do not pursue that here. ■

5. BOUNDS ON THE COMPLEXITY INDEX

We have shown that provided w is not too close to unity, then for fixed c, complexity depends only on the complexity index z. In this section we turn our attention to the complexity index.

Recall that $z = c/\lg p$. We begin our analysis of z by considering the cost per step, c. We distinguish between two kinds of problems.

We say a problem is explicit if the formula for f is given explicitly. For example, the calculation of $a^{1/2}$ by solving $f = x^2 - a$ is an explicit problem. The complexity of explicit problems has been studied by Paterson [72] and Kung [73a], [73b]. (Paterson and Kung take the efficiency index as basic.) We do not treat explicit problems here.

We say a problem is implicit if all we know about f are certain functionals of f. Classically the functionals are f and its derivatives evaluated at certain points. These functionals may be thought of as black boxes which deliver an output for any input. Kacewicz [75] has shown that integral functionals are of interest. The question of what functionals may be used in the solution of a problem are beyond the scope of this paper. We confine ourselves to implicit problems for the remainder of this paper.

We assume the same set of functionals is used at each step of the iteration. The set of functionals used by an iteration algorithm ϕ is called the information set $\mathfrak{N}$. Woźniakowski [75a] gives many examples of $\mathfrak{N}$. Let the *information complexity* $u = u(f,\mathfrak{N})$ be the cost of evaluating functionals on the information set $\mathfrak{N}$ and let the *combinatory complexity* $d = d(\phi)$ be the cost of combining functionals (see Kung and Traub [74b]). We assume that each arithmetic operation costs unity and denote the number of operations for one evaluation of $f^{(j)}$ by $c(f^{(j)})$. The following simple example may serve to illustrate the definition.

EXAMPLE 5.1. Let ϕ be Newton-Raphson iteration $x_{i+1} = \phi(x_i) = x_i = f(x_i)/f'(x_i)$, $i = 0,\ldots,k-1$. Then $\mathfrak{N} = \{f(x_i), f'(x_i)\}$, $u(f,\mathfrak{N}) = c(f) + c(f')$, $d(\phi) = 2$. □

Up to this point we have illustrated the concepts with algorithms. Computational complexity deals with classes of algorithms and we turn to our central concern, lower and upper bounds on classes of algorithms. As usual the difficult problem is obtaining lower bounds. Good lower bounds may be obtained from good lower bounds on cost and good upper bounds on order. The problem of maximal order is a difficult one about which a great deal has been recently learned (Meersman [75], Woźniakowski [75a], [75b]). Part of the mathematical difficulty of the subject deals with the problem of maximal order. Note however that maximal order does not necessarily minimize complexity; we deal with this in a future paper. Upper bounds are obtained from algorithms. An interesting question here is a good upper bound on the combinatory complexity of a class of algorithms. Brent and Kung [75] have obtained a surprising new upper bound, O(n lg n), on the combinatory complexity on a family of nth order one-point iterations based on inverse interpolation.

It is convenient to index our algorithms by n, the number of elements in the information set $\mathfrak{N}$. We illustrate the issues with two examples.

EXAMPLE 5.2. Let ϕ_n denote any one-point iteration with $\mathfrak{N} = \{f(x_i), f'(x_i), \ldots, f^{(n-1)}(x_i)\}$. Let $c_f = \min_i c(f^{(i)})$. Then $u(f,\mathfrak{N}) \geq nc_f$. For simplicity we use the linear lower bound $d(\phi_n) \geq n-1$. (No non-linear lower bound is known.) A sharp upper bound on the order of one-point iteration (Traub [64], Kung and Traub [74a]) is $p \leq n$. Hence

$$z(\phi_n, f) \geq \frac{nc_f+n-1}{\lg n}$$

$$z(\phi_n, f) \geq \frac{nc_f+n-1}{\lg n} \geq \frac{3c_f+2}{\lg 3}$$

provided only that $c_f \geq 4$ (Kung and Traub [74b]). Hence for any one-point iteration with $w_L \leq t$

$$(5.1) \quad \text{comp} \geq \frac{3c_f+2}{\lg 3}(\lg t - \lg\lg t).$$

On the other hand there exists a one-point iteration which uses f, f', f'' and such that p = 3. Hence if $w_U \geq 2$,

$$(5.2) \quad \text{comp} \le \frac{c(f)+c(f')+c(f'')}{\lg 3} \lg(1+t).$$

For problems such that $c(f) \simeq c(f') \simeq c(f'') \simeq c_f$ the lower and upper bounds of (5.1) and (5.2) are close together. ∎

EXAMPLE 5.3. Kung and Traub [74a] show there exists an iteration ψ_n for which the information set $\mathfrak{N}$ consists of n evaluation of f with $p(\psi_n) = 2^{n-1}$ and $d(\psi_n) = \frac{3}{2}n^2 + \frac{3}{2}n - 7$. Hence

$$z(\psi_n) = \frac{nc(f)+\frac{3}{2}n^2+\frac{3}{2}n-7}{n-1}.$$

The complexity index is minimzed (Kung and Traub [74b]) at $n^* = \text{round}[1 + (\frac{2}{3}(c(f)-4)^{1/2}] = O(c(f))^{1/2}$ and

$$z(\psi_{n^*}) = c(f)\Bigg/\left(1 + \frac{\xi}{(c(f))^{1/2}}\right), \xi > 0.$$

It would only be reasonable to use this high an order iteration for very small ε. Assume $t >> p^* = 2^{n-1}$.

Observe that $z(\psi_n)$ is a very "flat" function of n. Thus $z(\psi_3) = \frac{3}{2}c(f) + \frac{11}{2}$ and comparing this with $z(\psi_{n^*})$ shows we can gain only another $\frac{1}{2}c(f)$.

Let Φ denote the class of all multipoint iterations for which $w_U \ge 2$. Then

$$\text{comp}(\Phi) \le c(f)\lg(1+t)\Bigg/\left(1 + \frac{\xi}{(c(f))^{1/2}}\right). \quad ∎$$

We can obtain a lower bound on the complexity of the class of multipoint iterations by using an upper bound on the maximal order of any multipoint iteration and a lower bound on the combinatorial complexity. Kung and Traub [74a] conjecture that any iteration without memory which uses n pieces of information per step has order $p \le 2^{n-1}$. This conjecture seems difficult to prove in general (Woźniakowski [75b]) but has been established for many important cases (Kung and Traub [73], Meersman [75], Woźniakowski [75b]).

6. SUMMARY AND EXTENSIONS TO THE MODEL

We have constructed a non-asymptotic theory of iterative computational complexity with strict lower and upper bounds. In order to make the complexity ideas as accessible as possible we have limited ourselves to scalar non-linear problems. The natural setting for this work is in a Banach space of finite or infinite dimension and we shall do our analysis in this setting in a future paper. We have focussed on the simplified model $e_i = Ae_{i-1}^p$. More realistic models include some of the following features;

1. $e_i = A_i e_{i-1}^p$ under various assumptions on the structure of A_i.
2. $e_i = A_i e_{i-1}^{p_1} \cdots e_{i-m}^{p_m}$. This is the appropriate model for iterations with memory.
3. Variable cost per iteration, c_i.
4. Include round-off error. Then e_i will not converge to zero.

We plan to analyze these more realistic models in the future. We also intend to investigate additional basic properties of complexity. Our various results will be used to analyze the complexity of important problems in science and engineering.

ACKNOWLEDGMENT

We thank H. T. Kung for his comments on this paper.

REFERENCES

Borodin and Munro [75] Borodin, A. and Munro, I., The Computational Complexity of Algebraic and Numeric Problems, American Elsevier, 1975.

Brent and Kung [75] Brent, R. P. and Kung, H. T., to appear.

Kacewicz [75] Kacewicz, B., "The Use of Integrals in the Solution of Nonlinear Equations in N Dimensions," to appear in Analytic Computational Complexity, edited by J. F. Traub, Academic Press, 1975.

Kung [73a] Kung, H. T., "A Bound on the Multiplicative Efficiency of Iteration," Journal of Computer and System Sciences 7, 1973, 334-342.

Kung [73b] Kung, H. T., "The Computational Complexity of Algebraic Numbers," SIAM J. Numer. Anal. 12, 1975, 89-96.

Kung and Traub [74a] Kung, H. T. and Traub, J. F., "Optimal Order of One-Point and Multipoint Iteration," Journal of the Association for Computing Machinery 21, 1974, 643-651.

Kung and Traub [74b] Kung, H. T. and Traub, J. F., "Computational Complexity of One-Point and Multi-Point Iteration," Complexity of Computation, edited by R. Karp, American Mathematical Society, 1974, 149-160.

Meersman [75] Meersman, R., "Optimal Use of Information in Certain Iterative Processes," to appear in Analytic Computational Complexity, edited by J. F. Traub, Academic Press, 1975.

Paterson [72] Paterson, M. S., "Efficient Iterations for Algebraic Numbers," in Complexity of Computer Computations, edited by R. Miller and J. W. Thatcher, Plenum Press, New York, 1972. 41-52.

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

Traub [72] Traub, J. F., "Computational Complexity of Iterative Processes," SIAM Journal of Computing 1, 1972, 167-179.

Woźniakowski [75 a] Woźniakowski, H., "Generalized Information and Maximal Order of Iteration for Operator Equations," SIAM J. Numer. Anal. 12, 1975, 121-135.

Woźniakowski [75 b] Woźniakowski, H., "Maximal Order of Multipoint Iterations Using n Evaluations," to appear in Analytic Computational Complexity, edited by J. F. Traub, Academic Press, 1975.