MATHEMATICS OF COMPUTATION
VOLUME 38, NUMBER 157
JANUARY 1982

Fixed Point Iteration with Inexact Function Values

By Peter Alfeld

Abstract. In many iterative schemes, the precision of each step depends on the computational effort spent on that step. A method of specifying a suitable amount of computation at each step is described. The approach is adaptive and aimed at minimizing the overall computational cost subject to attaining a final iterate that satisfies a suitable error criterion. General and particular cost functions are considered, and a numerical example is given.

1. Introduction. The following situation is common in numerical analysis: Given is an iterative scheme in which each step itself can only be carried out approximately. The precision of each step depends on the amount of computational effort spent on that step. Normally, it would be advantageous to carry out the iteration steps with a very low precision (i.e., cheaply) initially and then to tighten the precision requirements as the iteration approaches the final approximation, only then increasing the cost of each iteration step.

To clarify the concepts, consider the following examples: *Iterating a series*: If a fixed point iteration function g, say, is given by an infinite series, then the computational effort spent at each step depends on the number of terms after which the series is truncated. The numerical example in Section 5 is of this type. *Nonlinear Equations*: Modified Newton methods in which the step is computed approximately in an inner iteration are a common source of fixed point iterations with inexact function values (see, e.g., [1], [5], [7]). *Unconstrained Minimization*: In those optimization algorithms involving linear searches (see, e.g., [4, pp. 7–8]), the computational effort at each step depends on the precision with which each linear search is carried out. *Constrained Optimization*: One approach to solving constrained optimization problems consists of transforming them into an infinite sequence of unconstrained problems, using barrier or penalty functions. (See, e.g., [6, Chapter 6].) Typically, each of the unconstrained problems is solved only approximately, using some numerical method. Note that in view of the preceding example we are iterating our basic concept. *Shooting Methods for Boundary Value Problems*: Often boundary value problems of ordinary differential equations are transformed into an infinite sequence of initial value problems (see, e.g., [3, pp. 262–264]), each of which has to be solved numerically, and therefore approximately.

In all of the above examples, it is apparent that computational efficiency could be gained by judiciously prescribing the computational effort to be spent at each step of the iteration.

Received July 8, 1980; revised June 12, 1981.
1980 *Mathematics Subject Classification.* Primary 65–00; Secondary 65B99, 65H10, 65K05.
Key words and phrases. Iteration, fixed point iteration, efficiency, numerical analysis.

0025-5718/82/0000-0471/$04.00

In this paper, we consider the special case of a linearly convergent fixed point iteration. No attempt has been made to phrase all of the above examples in this framework, but many of the concepts and results are applicable.

The basic approach consists of attempting to minimize the overall computational cost subject to attaining a bound on the error in the final iterate. The suggested numerical algorithm depends on quantities that can be estimated during the iteration, primarily the locally valid Lipschitz constant. Since this is dependent on the particular fixed point problem and may change as the iteration proceeds, the algorithm is designed to adapt itself to the particular problem and to changing circumstances.

The following sections contain: a formal description of the problem under consideration and some general results (Section 2), the analysis of a particular cost function (Section 3), computational aspects and a suggested algorithm (Section 4), a numerical example (Section 5), and conclusions.

2. Fixed Point Iteration. At the beginning of this section, we will introduce the type of iteration under consideration and define some of the terms that we will be using.

The traditional fixed point iteration is defined by

$$x_{n+1} = G(x_n), \qquad n = 0, 1, 2, \ldots, \tag{2.1}$$

where G: $\mathbf{R}^d \to \mathbf{R}^d$ is a given function and x_0 is a given initial vector.

In this paper, we consider instead functions

$$g\colon \mathbf{R}^d \times [0, \infty) \to \mathbf{R}^d$$

and iterations of the form

$$x_0 \in \mathbf{R}^d \text{ given}, \qquad x_{n+1} = g(x_n, \varepsilon_n), \qquad n = 0, 1, \ldots, N-1. \tag{2.2}$$

We are interested in the convergence of (2.2) to a point $\overset{*}{x}$ satisfying

$$\overset{*}{x} = g(\overset{*}{x}, 0). \tag{2.3}$$

We will refer to $\overset{*}{x}$ simply as a *fixed point* of g.

For all $x \in \mathbf{R}^d$ and $\varepsilon \geqslant 0$ the vector $g(x, \varepsilon)$ is an approximation of $g(x, 0)$. We assume that for given $\varepsilon > 0$ we can carry out the approximation such that

$$\| g(x, \varepsilon) - g(x, 0)\| \leqslant \varepsilon. \tag{2.4}$$

The norm $\| \cdot \|$ is arbitrary, but fixed throughout this paper.

The numbers $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}$ are called the *control variables* of the iteration (2.2). It is convenient to collect them into a *control vector* $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]^T$. We will always assume that all control variables are nonnegative.

The iteration (2.2) is said to be *infinitely precise* if all control variables are zero. Thus the traditional fixed point iteration (2.1) can be considered the special case of an infinitely precise iteration (2.2). Whenever it is convenient, we will identify $G(x)$ with $g(x, 0)$.

We will refer to the quantity $\|x_N - \overset{*}{x}\|$ as the *final error*. Thus we distinguish two types of accuracy: that with respect to an approximation of $\overset{*}{x}$, described by the term *final error*, and that with respect to an approximation of $g(x_n, 0)$, described by the term *precision*.

The number N in (2.2) will be determined so as to keep the final error below a certain bound. It will not be known at the beginning of the iteration, but rather will be estimated and reestimated as the iteration proceeds. We refer to N as the *target number*.

It is well known (by the Contraction-Mapping Theorem) that the traditional iteration (2.1) converges if there exists a Lipschitz constant $L < 1$ satisfying $\|G(x) - G(y)\| \leqslant L\|x - y\|$ for all $x, y \in \mathbf{R}^d$.

We make the corresponding assumption that there exists a Lipschitz constant $L < 1$ satisfying

$$\|g(x, 0) - g(y, 0)\| \leqslant L\|x - y\| \tag{2.5}$$

for all $x, y \in \mathbf{R}^d$.

Note our assumption that both the relations (2.4) and (2.5) are valid globally, i.e., for all $x, y \in \mathbf{R}^d$. This assumption is for convenience only and can be slackened considerably. However, in this paper we concentrate on the optimal choice of the control variables $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}$. It is a consequence of the global validity of (2.5) that g possesses a unique fixed point $\overset{*}{x}$ satisfying (2.3).

We assume that the computational effort of evaluating $g(x, \varepsilon)$ is given by a *cost function* c: $(0, \infty) \to \mathbf{R}$ satisfying

$$c(\varepsilon) \geqslant 0 \quad \text{and} \quad \varepsilon < \bar{\varepsilon} \Rightarrow c(\varepsilon) \geqslant c(\bar{\varepsilon}) \tag{2.6}$$

for all $\varepsilon, \bar{\varepsilon} > 0$. Thus we assume that the cost of evaluating $g(x, \varepsilon)$ is positive and monotonic decreasing with ε. This is reasonable. More severe (and unrealistic in some cases) is the simplifying assumption that the cost is independent of x.

At this stage, we have introduced the language to be used in the sequel. The basic plan is to derive a bound on the final error (Theorems 1 and 2) and to give necessary and sufficient conditions for a target number and a control vector to minimize the overall numerical cost subject to keeping that bound below a specified value (Theorem 5). Theorems 3 and 4 deal with general features of the iteration (2.2) and are not immediately relevant to the optimization problem. They can be skipped without loss of continuity.

THEOREM 1. *Let $x_0 \in \mathbf{R}^d$ be given, and let x_N be defined by* (2.2) *for some target number N and some control vector $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]^T$. Then*

$$\|x_N - \overset{*}{x}\| \leqslant \sum_{n=0}^{N-1} L^{N-1-n}\varepsilon_n + L^N\|x_0 - \overset{*}{x}\|. \tag{2.7}$$

Proof. The proof is by induction in N and straightforward, using (2.4) and (2.5). □

The bound given in (2.7) has the disadvantage of the term $\|x_0 - \overset{*}{x}\|$ normally being unavailable. The next theorem gives a cruder bound that can be computed (or estimated) if L is known (or can be estimated).

THEOREM 2. *Under the assumptions of Theorem* 1 *there holds*

$$\|x_N - \overset{*}{x}\| \leqslant \sum_{n=0}^{N-1} L^{N-1-n}\varepsilon_n + \frac{L^N}{1 - L}(\|x_1 - x_0\| + \varepsilon_0). \tag{2.8}$$

Proof. Theorem 2 follows from Theorem 1 by observing that

$$\begin{aligned}\|x_0 - \overset{*}{x}\| &\leqslant \|x_0 - g(x_0, \varepsilon_0)\| + \|g(x_0, \varepsilon_0) - g(x_0, 0)\| \\ &\quad + \|g(x_0, 0) - g(\overset{*}{x}, 0)\| + \|g(\overset{*}{x}, 0) - \overset{*}{x}\| \\ &\leqslant \|x_1 - x_0\| + \varepsilon_0 + L\|x_0 - \overset{*}{x}\|,\end{aligned}$$

which implies that

$$\|x_0 - \overset{*}{x}\| \leqslant \frac{1}{1-L}(\varepsilon_0 + \|x_1 - x_0\|). \quad \square$$

Independent of the optimization problem we are considering, it is interesting to ask under which conditions on the control variables the iteration (2.2) will converge to $\overset{*}{x}$ as we let the target number N tend to infinity. This question cannot be answered without further knowledge about g. For example, if $g(x, \varepsilon) = \overset{*}{x}$ for all $x \in \mathbf{R}^d$ and $\varepsilon \geqslant 0$, then the iteration will converge for any choice of the control variables. On the other hand, if $g(x, \varepsilon) = \overset{*}{x} + \varepsilon v$, where v is any vector satisfying $\|v\| = 1$, then we have to require that $\lim_{n\to\infty} \varepsilon_n = 0$. In both cases the relation (2.4) is satisfied.

A slightly less interesting but more tractable question to ask is when the bounds given in Theorems 1 and 2 will converge to zero as N tends to infinity. Obviously, the final error will tend to zero if the bounds tend to zero. The following theorem gives a necessary and sufficient condition.

THEOREM 3. *For given $x_0 \in \mathbf{R}^d$, $L < 1$ and control variables $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$, the right-hand sides of* (2.7) *and* (2.8) *converge to zero as N tends to infinity if and only if*

$$\lim_{n\to\infty} \varepsilon_n = 0. \tag{2.9}$$

Proof. Since $L < 1$, for both (2.7) and (2.8) we only have to show that

$$\lim_{N\to\infty} \sum_{n=0}^{N-1} L^{N-1-n}\varepsilon_n = 0 \Leftrightarrow \lim_{n\to\infty} \varepsilon_n = 0.$$

Assume that $\lim_{N\to\infty} \sum_{n=0}^{N-1} L^{N-1-n}\varepsilon_n = 0$ and that the sequence $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$ does not converge to zero. Then there exists a constant $\gamma > 0$ and a sequence of integers $n_1, n_2, n_3, \ldots$ such that

$$\varepsilon_{n_i} > \gamma > 0 \quad \text{for } i = 0, 1, 2, \ldots.$$

(Recall that we assume the control variables $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$ to be nonnegative.)

It follows for all integers k that

$$\sum_{n_i < k} L^{k-n_i}\varepsilon_{n_i} \geqslant \gamma \sum_{n_i < k} L^{k-n_i} = \gamma B_k,$$

where $B_k = \sum_{n_i < k} L^{k-n_i}$. Thus, for all $r = 0, 1, 2, \ldots$, $B_{n_r} \geqslant 1$ and

$$\sum_{i=0}^{n_r} L^{n_r-i}\varepsilon_i \geqslant \gamma \sum_{i=0}^{r} L^{n_r-n_i} \geqslant \gamma > 0,$$

which is a contradiction, establishing the "only if" part of the theorem.

Assume now that $\lim_{n\to\infty} \varepsilon_n = 0$, and assume we are given some $\varepsilon > 0$. Then there exists an integer M such that $\varepsilon_n < \frac{1}{2}(1 - L)\varepsilon$ for all $n > M$. Also, there exists an integer $\overline{M} \geqslant M$ such that

$$L^{k-M} \sum_{i=0}^{M} L^{M-i}\varepsilon_i < \frac{\varepsilon}{2}$$

for all $k > \overline{M}$. Hence, for all $k > \overline{M}$,

$$\begin{aligned} \sum_{i=0}^{k} L^{k-i}\varepsilon_i &\leqslant L^{k-M} \sum_{i=0}^{M} L^{M-i}\varepsilon_i + \sum_{i=M+1}^{k} L^{k-i}\varepsilon_i \\ &\leqslant \frac{\varepsilon}{2} + \frac{1}{2}(1 - L)\varepsilon \sum_{i=0}^{\infty} L^i = \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon, \end{aligned}$$

which completes the proof. □

Remark. It is plausible that $\lim_{n\to 0} \varepsilon_n = 0$ is a necessary condition for the bounds in Theorem 1 to converge to zero, but it is not obvious that this condition is also sufficient.

We now start considering the overall cost of the iteration (2.2) defined by a control vector $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]$. This is given by

$$\Psi(N, e) := \sum_{n=0}^{N-1} c(\varepsilon_n). \tag{2.10}$$

Ideally, we would like to find N and e so as to minimize Ψ subject to keeping the final error $\|x_N - \overset{*}{x}\|$ below a specified number η, say. This is impractical, but it is possible to minimize Ψ subject to keeping one of the bounds given in Theorems 1 and 2 below η. In view of its computability, the bound (2.8) given in Theorem 2 might seem more attractive to use, but it has certain shortcomings that will be pointed out below. We will base our analysis on the bound (2.7) and deal with the problem of estimating $\|x_0 - \overset{*}{x}\|$ in Section 4.

Thus we arrive at the constrained minimization problem (COP):

Find N and $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]^T$ such that

$$\Psi(N, e) = \sum_{n=0}^{N-1} c(\varepsilon_n) = \min \tag{2.11}$$

subject to

$$\Phi(N, e) := \sum_{n=0}^{N-1} L^{N-1-n}\varepsilon_n + L^N\|x_0 - \overset{*}{x}\| = \eta. \tag{2.12}$$

Note that in (2.12) we assume equality instead of inequality. This simplifies the subsequent analysis and is no restriction of generality, since the monotonicity of c implies that there exist minimizing N and e for which $\phi(N, e) = \eta$ whenever COP has a solution.

Before we tackle the above problem we take a formal note of the intuitively obvious fact that the optimum sequence of control variables is monotonically decreasing.

THEOREM 4. *Assume* $L \in (0, 1)$, $N \in \mathbf{N}$, *and* $\eta \in \mathbf{R}$ *are given constants, and* $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]^T$ *minimizes* $\psi(N, e)$ *subject to* $\Phi(N, e) \leqslant \eta$. *Also assume that* c *is strictly monotonically decreasing. Then, for all* $n = 1, 2, \ldots, N-1$,

$$\varepsilon_n \leqslant \varepsilon_{n-1}.$$

Proof. The proof is by contradiction. First observe that because of the strict monotonicity of c, the constraint $\Phi(N, e) \leqslant \eta$ is active, i.e., at the minimizing (N, e) we have $\Phi(N, e) = \eta$. Now assume that for some $n \in \{1, 2, \ldots, N-1\}$ there holds $\varepsilon_n > \varepsilon_{n-1}$. Let $\tilde{e} = [\tilde{\varepsilon}_0, \tilde{\varepsilon}_1, \ldots, \tilde{\varepsilon}_{N-1}]^T$, where $\tilde{\varepsilon}_i = \varepsilon_i$ if $i \notin \{n, n-1\}$, $\tilde{\varepsilon}_n = \varepsilon_{n-1}$, $\tilde{\varepsilon}_{n-1} = \varepsilon_n$. Then obviously $\Psi(N, e) = \Psi(N, \tilde{e})$. Furthermore

$$\begin{aligned}\Phi(N, e) - \Phi(N, \tilde{e}) &= L^{N-1-n}(\varepsilon_n - \tilde{\varepsilon}_n) + L^{N-n}(\varepsilon_{n-1} - \tilde{\varepsilon}_{n-1}) \\ &= (\varepsilon_n - \varepsilon_{n-1})L^{N-1-n}(1 - L) > 0.\end{aligned}$$

Hence $\Phi(N, \tilde{e}) < \Phi(N, e) = \eta$; i.e., $\psi(N, e) = \psi(N, \tilde{e})$ without the constraint being active. Hence the value of $\psi(N, e)$ can be reduced, which is a contradiction. □

Remark. It is easy to see that, in the case $L \in (0, \frac{1}{2})$, ε_0 has to be less than ε_1 if the cost is to be minimized subject to keeping the bound given in Theorem 2 below η. This is an artifact of bounding $\|x_0 - \overset{*}{x}\|$ by $(\|x_1 - x_0\| + \varepsilon_0)/(1 - L)$ (thus making the second term in (2.8) dependent on ε_0) and suggests basing the analysis and the algorithm on the more natural bound (2.7).

We now state necessary and sufficient conditions for the solution of COP.

THEOREM 5. *Assume* $N \in \mathbf{N}$ *and* $\eta > 0$ *are given, and* $c \in C^1(0, \infty)$. *Then* $e = [\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{N-1}]^T$ *uniquely minimizes* $\Psi(N, e)$ *subject to* $\Phi(N, e) = \eta$ *only if there exists a constant* λ *such that*

$$c'(\varepsilon_n) = \lambda L^{N-1-n}, \qquad n = 0, 1, \ldots, N-1. \tag{2.13}$$

If c *is strictly convex, then this condition is also sufficient.*

Proof. Consider the Lagrange function $L(\lambda, N, e) = \Psi(N, e) - \lambda\Phi(N, e)$, where λ is the Lagrange multiplier. A necessary condition for e to minimize $\Psi(N, e)$ subject to $\Phi(N, e) = \eta$ is that

$$\frac{\partial}{\partial e} L(\lambda, N, e) = 0$$

for some λ, which is just (2.13). Assume now that (2.13) is satisfied for some vector e. Since $\Phi(N, \bar{e})$ is linear in $\bar{e}$, the set $\{\bar{e}, \phi(N, \bar{e}) = \eta\}$ is affine (and convex). The function $\psi(N, e)$ is strictly convex in e (since $c(\varepsilon)$ is a strictly convex function of ε). Thus the stationary point e is a local minimizer of $\psi(N, \cdot)$. Since a strictly convex function possesses at most one local minimizer in a convex set, e is unique. This completes the proof. □

We have now assembled the basic tools to solve COP. For any particular cost function c, we can proceed as follows:

1. Solve (2.13) for ε_n in terms of n, λ, L, and N.
2. Compute $\Phi(N, e)$ and solve (2.12) for λ (in terms of L and N).
3. Minimize $\Psi(N, e)$ with respect to N. Given L, this is an unconstrained minimization problem.

3. A Particular Cost Function. We will now study a particular instance of the cost function c. Note that the solution of COP is independent of any constant factor multiplying c. We can thus restrict our attention to some canonical form of c. For convenience, and because it will turn out to be reasonable computationally, we will treat the target number N as a *real* variable. The following theorem covers the case that $c(\varepsilon) = \varepsilon^{-p}$ for some $p > 0$.

THEOREM 6. *Suppose* $c(\varepsilon) = \varepsilon^{-p}$ *for some* $p > 0$. *Then the solution of the problem COP is given by*

$$N = -(p+1)\frac{\ln\|x_0 - \overset{*}{x}\|/\eta}{\ln L} \tag{3.1}$$

and

$$\varepsilon_n = \kappa L^{(1+n-N)/(p+1)}, \tag{3.2}$$

where

$$\kappa = \frac{1-\hat{L}}{1-\hat{L}^N}\left(\eta - L^N\|\overset{*}{x} - x_0\|\right) \tag{3.3}$$

and

$$\hat{L} = L^{p/(p+1)}. \tag{3.4}$$

Proof. The cost function c is continuously differentiable and strictly convex. Hence, by Theorem 5, for any fixed N, there exists a unique solution of COP which is characterized by (2.13). We will now follow the strategy outlined at the end of the preceding section.

The equation (2.13) becomes

$$-p\varepsilon^{-p-1} = \lambda L^{N-1-n},$$

which yields (3.2) with

$$\kappa = \left(-\frac{\lambda}{p}\right)^{-1/(p+1)}. \tag{3.5}$$

To determine κ (and thereby λ) we compute

$$\begin{aligned}\Phi(N, e) &= \sum_{n=0}^{N-1} L^{N-1-n}\kappa L^{(1+n-N)/(p+1)} + L^N\|x_0 - \overset{*}{x}\| \\ &= \kappa\frac{\hat{L}^N - 1}{\hat{L} - 1} + L^N\|x_0 - \overset{*}{x}\|,\end{aligned}$$

where $\hat{L}$ is given by (3.4). Solving $\Phi(N, e) = \eta$ for κ yields (3.3).

To obtain the optimum (real) value of N, we compute

$$\Psi(N, e) = \sum_{n=0}^{N-1} \varepsilon_n^{-p} = \kappa^{-p}\frac{\hat{L}^{-1} - \hat{L}^{N-1}}{\hat{L}^{-1} - 1}. \tag{3.6}$$

Differentiating w.r.t. N and setting equal to zero yields, after some manipulation, the equation

$$\begin{aligned}\left(\eta - L^N\|x_0 - \overset{*}{x}\|\right)^{-(p+1)}(\hat{L}^N - 1)^p p(1-\hat{L})^{-(p+1)}(\ln L) \\ \times \left(\eta\hat{L}^N - L^N\|x_0 - \overset{*}{x}\|\right) = 0.\end{aligned} \tag{3.7}$$

Of the factors on the left-hand side of (3.7), the first five are nonzero, independent of the choice of $N > 0$. The unique value of N that makes the last factor vanish is given by (3.1). That this value of N indeed minimizes Ψ follows from its uniqueness, the fact that Ψ is bounded below, and the observation that the cost subject to satisfying (2.12) tends to infinity as N tends to

$$N_0 = -\frac{\ln\|x_0 - \overset{*}{x}\|/\eta}{\ln L} \tag{3.8}$$

(see Remark 2 below). □

Remarks. 1. The Lagrange multiplier λ and the actual cost of the optimal iteration can be computed using (3.5) and (3.6), respectively.

2. In the case of an infinitely precise iteration (2.2) (i.e. $\varepsilon_n = 0$, $n = 0, 1, \ldots, N - 1$), the number of iterations required subject to (2.12) is given by N_0 defined in (3.8). By (3.1), the optimal number of iterations is $(p + 1)N_0$. This is a surprisingly simple result.

4. Computational Aspects. In this section, we discuss how the results in the preceding sections can be used to solve iterations of the type (2.2) efficiently. No effort has been made, however, to specify a robust algorithm to the point that it can be readily implemented into a piece of production software.

We will assume that the actual cost can be modelled by a function $c(\varepsilon) = \varepsilon^{-p}$, for some given fixed $p > 0$, and apply Theorem 6. (Later we will discuss the validity of this approach and a method of approximating p.)

Any algorithm should be such that the user has to supply as little information about the fixed point problem as possible, i.e., that as many of the important parameters as possible are generated automatically. Moreover, the nature of the problem may change as the iteration proceeds, thus making it desirable to update the controlling parameters as soon as new information becomes available. Hence we will consider each point x_k as the starting point of a new iteration (2.2), (i.e., x_k will play the role of x_0). For each x_k we will reestimate the key quantities L and $\|x_k - \overset{*}{x}\|$, based on which we will compute a new estimate of N. Note that, since the optimal value of N cannot be computed exactly, there is no point in insisting on N being integer.

It is important to note that we need estimates of L and $\|x_k - \overset{*}{x}\|$ rather than bounds. If $\|x_k - \overset{*}{x}\|$ is overestimated, the algorithm will choose ε_k too large, and the iteration may never converge. Likewise, if L is overestimated, the target number and therefore ε_k will be too large. On the other hand, if $\|x_k - \overset{*}{x}\|$ or L are underestimated, then ε_k will be smaller than necessary, thus impairing efficiency.

In what follows, L will be approximated by $\tilde{L}$, $\|x_k - \overset{*}{x}\|$ by α, and N by $\tilde{N}$. A simple way of estimating L is given by

$$\tilde{L} = \frac{\|x_k - x_{k-1}\|}{\|x_{k-1} - x_{k-2}\|}.$$

However, due to the imprecision of (2.2), $\tilde{L}$ may be greater than 1. Thus we define

$$\tilde{L} = \min\left\{ \frac{\|x_k - x_{k-1}\|}{\|x_{k-1} - x_{k-2}\|}, \beta \right\}, \tag{4.1}$$

where $\beta \in (0, 1)$ is a constant that is either supplied by the user or treated as an internal parameter.

More difficult is the specification of α. Consider for the moment the case of a scalar iteration (2.1). Ignoring the imprecision of the iteration, it is easy to see that

$$|x_k - \overset{*}{x}| \leqslant \frac{L}{1 - L}|x_k - x_{k-1}|$$

suggesting

$$\alpha = \frac{L}{1 - L}\|x_k - x_{k-1}\|.$$

This works quite well if $G'(x_k) > 0$, and L is close to $G'(x_k)$. It is a gross overestimate, however, if $G'(x_k) < 0$. Therefore we consider, instead of (2.1), the iteration

$$x_{n+2} = G(G(x_k)).$$

The derivative of this iteration function is usually positive, and its Lipschitz constant is L^2, suggesting

$$\alpha = \frac{L^2}{1 - L^2}\|x_k - x_{k-2}\|, \tag{4.2}$$

which is a good estimate independent of the sign of G'.

In the system case we need to consider instead the eigenvalues of G'. If these have nonvanishing imaginary parts, then there will be components of the iterates that oscillate with various frequencies. In that case it is necessary to estimate the frequencies of the dominant eigenvalues or to monitor the behavior of the ϵ_n and force their convergence to zero (see Theorem 3). We will not pursue this question further and adopt instead (4.2).

Both the estimates (4.1) and (4.2) lead to a starting problem. They are not available if $k < 2$. Thus we require starting values of $\tilde{L}$ and α. Again, these may be internal parameters or be supplied by the user. Both L and $\|x_0 - \overset{*}{x}\|$ should be underestimated initially, rather than overestimated. Otherwise the precision of the first two steps would be too low, yielding unreliable values of $\tilde{L}$ and α at the following steps.

Given α and $\tilde{L}$, $\tilde{N}$ and ϵ_k can be computed from (3.1) and (3.2). $\tilde{N}$ is the estimated number of iteration steps still to go (hence the term target number). Thus it would be natural to terminate the iteration as soon as $\tilde{N} < 0$. However, it is safer to require that $N < -\delta < 0$ for some internal parameter $\delta > 0$.

The following outline provides a basis on which an algorithm can be built. The iteration counter is k.

Constant p algorithm

GIVEN x_0, η, α, $\tilde{L}$, β, p, δ; $k := 0$

REPEAT

IF $k > 1$ *THEN*

$$\tilde{L} := \min\left\{ \frac{\|x_k - x_{k-1}\|}{\|x_{k-1} - x_{k-2}\|}, \beta \right\}$$

$$AND \quad \alpha := \frac{L^2}{1 - L^2}\|x_k - x_{k-2}\|$$

$$\tilde{N} := -(p+1)\frac{\ln(\alpha/n)}{\ln \tilde{L}}$$

IF $\tilde{N} < -\delta$ *THEN* STOP

$$\bar{L} := \tilde{L}^{p/(p+1)}$$

$$\kappa := \frac{1 - \bar{L}}{1 - \bar{L}^N}\left(\eta - \tilde{L}^{\tilde{N}}\alpha\right)$$

$$\varepsilon_k := \kappa \tilde{L}^{(1-\tilde{N})/(p+1)}$$

$$x_{k+1} := g(x_k, \varepsilon_k)$$

$$k := k + 1$$

GO TO REPEAT

A weakness of the above approach is that p is constant and assumed to be known. Often cost functions will be integer valued step functions (giving, e.g., the number of evaluations or arithmetic operations) and certainly not of the form $c(\varepsilon) = \varepsilon^{-p}$ for any value p. One possibility of estimating p is local interpolation of the actual cost. Suppose we measure $c_{k-2} = c(\varepsilon_{k-2})$ and $c_{k-1} = c(\varepsilon_{k-1})$. Then solving $c_i = \gamma\varepsilon_i^{-p}$ $(i = k-2, k-1)$ for p yields

$$p = -\frac{\ln c_{k-2}/c_{k-1}}{\ln \varepsilon_{k-2}/\varepsilon_{k-1}}.$$

If c is not sufficiently smooth, least square approximation could be used. In numerical experiments the efficiency of the algorithm seemed to be very insensitive with respect to p.

5. Numerical Example. We consider the fixed point function

$$g(x, 0) = \gamma\left(\frac{2\pi^2}{3} - 4\sum_{k=1}^{\infty}\frac{(-1)^k}{k^2}\cos kx\right), \tag{5.1}$$

where γ is a parameter. This is just the Fourier series of $g(x, 0) = \gamma(\pi^2 - x^2)$. It turns out that

$$\overset{*}{x} = \frac{1}{2\gamma}\left(\sqrt{4\gamma^2\pi^2 + 1} - 1\right) \quad \text{and} \quad G'(\overset{*}{x}) = -2\gamma\overset{*}{x} < 0.$$

The series in (5.1) was truncated as soon as (2.4) was satisfied. The cost $c(\varepsilon)$ was defined to be the number of cos evaluations in (5.1). This generates a piecewise constant cost function with steps of greatly varying length.

The Constant p Algorithm was applied with $\beta = 0.99$, $\delta = 1$, $x_0 = \overset{*}{x} + 0.5$, $\eta = 10^{-7}$, and $\tilde{L} = \alpha = 0.1$ initially.

The following table gives numerical results for $\gamma = 0.2$ $(G'(\overset{*}{x}) = -0.61)$ and $\gamma = 0.25$ $(G'(\overset{*}{x}) = -0.86)$, and $p = 0.1, 0.2, \ldots, 1.0$. The integer m is the overall computational effort until convergence was reached. For comparison purposes the iteration was also carried out with ε_k kept constant and equal to the control variable at the last step of the Constant p Algorithm. The number a is the ratio of the effort for the constant precision iteration and m. We observe that the gain in efficiency is quite insensitive with respect to p (and γ). The actual efficiency is also insensitive with respect to p if $p > 0.3$. It is remarkable that the above results were obtained in spite of the true cost being a step function that is only poorly modelled by $c(\varepsilon) = \varepsilon_1^{-p}$, for any p. For all iterations the final error was smaller than the specified value of η. The computed value of $\tilde{L}$ was always less than β.

TABLE 1. *Numerical Results*

γ:	0.2		0.25	
p	m	a	m	a
0.1	4,776	5.0	25,761	5.1
0.2	4,061	5.4	19,984	5.2
0.3	3,723	5.6	18,013	5.3
0.4	2,826	6.1	13,816	5.0
0.5	2,826	5.8	13,117	5.9
0.6	3,077	5.0	12,852	6.0
0.7	2,834	6.0	11,966	5.9
0.8	2,833	5.9	12,494	5.5
0.9	2,943	4.8	12,243	5.4
1.0	2,943	4.7	12,130	5.4

Conclusions. A method has been suggested for carrying out inexact function iterations efficiently. Apparently this problem has not been tackled before in the generality attempted here. The ideas and concepts are applicable to a wide range of numerical problems, including the iteration of series, the solution of nonlinear equations, constrained and unconstrained minimization, and shooting methods for boundary value problems.

Acknowledgements. The technical manipulations in this paper were carried out using the symbol manipulation language REDUCE [2]. The author greatly appreciates the careful work of the referee who detected a significant error in the analysis of another particular cost function. Helpful discussions with Frank Stenger and Jim Keener of the University of Utah are gratefully acknowledged.

Department of Mathematics
University of Utah
Salt Lake City, Utah 84112

1. R. S. DEMBO, S. C. EISENSTAT & T. STEIHAUG, *Inexact Newton Methods*, Working Paper #47 (Series B), Yale School of Organization and Management, 1980.

2. A. C. HEARN, REDUCE *User's Manual*, 2nd ed., Report UCP-19, Department of Computer Science, University of Utah, 1973.

3. J. D. LAMBERT, *Computational Methods in Ordinary Differential Equations*, Wiley, New York, 1973.

4. W. MURRAY, *Numerical Methods for Unconstrained Optimization*, Academic Press, New York, 1972.

→ V. PEREYRA, "Accelerating the convergence of discretization algorithms," *SIAM J. Numer. Anal.*, v. 4, 1967, pp. 508–533.

6. D. M. RYAN, "Penalty and barrier functions," in *Numerical Methods for Constrained Optimization* (P. E. Gill and W. Murray, Eds.), Academic Press, New York, 1974.

7. A. H. SHERMAN, "On Newton-iterative methods for the solution of systems of nonlinear equations," *SIAM J. Numer. Anal.*, v. 15, 1978, pp. 755–771.