Interpreting Rate Analysis

Harvey J. Greenberg,
Management Science Systems

Since rate analysis has been discovered by mathematical programmers, it has been misused in recommending one algorithm over another. The primary purpose of this note is to demonstrate this assertion and provide some insight into interpretative rate analysis. Further, I shall describe important uses of rate analysis, namely for acceleration and design of cut-off rules.

Let \( \{s_k\} \) be a scalar sequence converging to zero. Rate analysis is concerned with describing the asymptotic form of convergence, especially for the purpose of measuring "speed". Traditionally the order of convergence is defined to be [1]:

\[
p \equiv \text{Sup} \{q: \lim_{k \to \infty} \frac{s_k^{q+1}}{s_k^{q}} < \infty \}.
\]

Notice that the limit superior is used, resulting in a rather conservative measure.

In general, it is possible to have any nonnegative order, but when the sequence is nonincreasing, then the order is at least 1. The cases \( p = 1 \) and \( p = 2 \) arise frequently in mathematical programming, and they are called linear and quadratic orders of convergence, respectively.

For order \( p \) the rate ratio is defined to be:

\[
r \equiv \lim_{k \to \infty} \left( \frac{s_k^{p+1}}{s_k^p} \right).
\]

Thus, order \( (p) \) and rate \( (r) \) are used to describe asymptotic behavior of a sequence.

Of special interest is the case of linear order with \( r = 0 \). This is called superlinear convergence. Similarly, if \( p = 1 \) and \( r = 1 \), this is called sublinear convergence.

The term "sublinear" highlights a minor confusion in terminology. A geometric sequence, of the form

\[
s^k = s_0 r^k,
\]

where \( |r| < 1 \), is of linear order (with rate, \( r \)). However, "geometric convergence" is not synonymous with "linear convergence."
An example of sublinear convergence is the arithmetic sequence, $s_k = 1/k$. Superlinear convergence is exemplified by $s_k = (1/k)^k$.

In mathematical programming we conceptualize an operand sequence of the form $\{A^kx\}$, where $A$ is the algorithm map (see Zangwill [2]). Two associated sequences are the policy error, $||A^kx - x^\infty||$ and the value, $|zA^kx - zx^\infty|$, where we suppose $\{A^kx\} + x^\infty$ and $z(.)$ is an improvement function (e.g., objective value). In what immediately follows it is not important what $\{s_k\}$ actually represents, and the above two (noncomputable) sequences are merely two important examples.

Here are some myths I wish to dispell:

I. If algorithm A has higher order than algorithm B, then A is preferred.

II. If algorithms A and B have the same order, the one with lower rate ratio is preferred.

III. If algorithm A is at least N-step superlinearly convergent, while B may not be, then algorithm A is preferred.

These three assertions, which I label "myths", are related and appear in many forms in the nonlinear programming literature.

One argument to dispell them is to note the dependence on the work-per-iteration; this leads to a modified measure called "efficiency." Alternatively, one can argue that machine error may be an increasing function of order (e.g., computing derivatives) and hence may cause poorer performance. A pragmatic challenge is whether we can afford to reach "asymptotic" behavior anyway.

These are some of the arguments against the evaluative use of rate analysis, but none of these really reaches the heart of the matter, namely the basic definitions of order and rate.

**Proposition**: Given any null sequence, $\{s_k\}$, there exists another sequence, $\{t_k\}$, such that

$t_k < s_k$ for all $k$.

and $\{t_k\}$ is sublinearly convergent.

**Proof**: Define

$$t_k = \begin{cases} s_k & \text{if } k \text{ is even} \\ u_k & \text{if } k \text{ is odd} \end{cases}$$

By choosing $\{ u_k \}$ (say equal to $2^{-k}$) we can force $\{t_k\}$ to converge to zero arbitrarily "fast," certainly faster than $\{s_k\}$. However, due to the constant plateaux at odd iterations the convergence is sublinear.

A little thought should illuminate the caution that must be exercised when using rate analysis as a performance yardstick.
Brent [1] describes alternative definitions; some offer an improvement. For example, one may define

\[ r = \inf_{n \geq 1} \lim_{k \to \infty} \left( \frac{s_{k+n}}{s_k} \right) \cdot \frac{1}{n} \]

This enables an averaging, say if \( s_k \) is constant for \( k = 1, 2, \ldots, n-1 \) and then suddenly realizes a large gain. In that case this rate would be equivalent to a linear sequence whose values steadily converge. (Note that here \( n \)-step and ordinary convergence become indistinguishable, which is quite appropriate.)

The example used in the proposition can be extended so that \( \{t_k\} \) remains constant over plateaux of increasing duration (and hence exceeds any constant \( n \)) to still yield sublinear convergence despite its being uniformly closer to the limit (zero).

Of course, the situation is even worse. The above proposition raises serious questions, assuming we could deduce exact rates for two algorithms. This is rarely the case. One may then question the value of rate analysis and how we could assimilate the knowledge of asymptotic behaviour.

Jones [3] has cast numerical extrapolation theory into an interesting light which views the base algorithm as an information generator, partitioned into signal and noise. At the risk of oversimplifying, let me briefly explain part of the pertinent idea.

Suppose we know

\[ A^{k+1} x = F(A^k x) , \]

as for example in linear convergence where

\[ A^{k+1} x = T A^k x + S . \]

Further, suppose one application of \( A \) is more costly than obtaining a fixed point of \( F \). Then, an extrapolation of \( \{A^k x\} \) is to use the history to estimate the functional parameters (e.g., \( T \) and \( S \)) followed by obtaining

\[ \hat{x} = F(\hat{x}) . \]

For example, linear convergence leads to finding \( \hat{x} \) which solves

\[ (I-T) \hat{x} = S \]

Jones suggests restricting \( T \) to be diagonal, so each coordinate is viewed as a scalar sequence. Then, with least squares estimation one can obtain \( \hat{x} \) rather easily. (Jones also noted that if only the latest three points are used in the estimate, then the procedure is Aitken's \( \delta^2 \)-method.)
The point to be made here is summarized with the following:

**Proposition:** It may be cheaper to accelerate a slow algorithm whose nature of slowness is known precisely than to apply a fast algorithm whose rate is not readily determined.

A second use of rate analysis is the design of cut-off rules [4]. This is exemplified by Porteus' [5] modified value iteration algorithm in dynamic programming. Basically, Porteus first noted the best known error bound for ordinary value iteration; call it $e_k$. Then, his algorithm was shown to have bound $p_k$, where $p_k < e_k$.

The work per iteration is roughly the same, so Porteus deduced his algorithm is better. Is this valid?

I claim it is if the stopping rule used in each case has the form:

STOP if error bound $< \epsilon$.

Then, even though value iteration may produce a better sequence (Porteus does not rule this out), the modified procedure stops sooner, and hence the total computing cost is less.

This can be summarized with the following:

**Proposition:** If two algorithms require the same work per iteration and one produces a state sequence uniformly closer to a stopping state, then it is preferred.

In summary, I claim that rate analysis has been misused in evaluating algorithms for the purpose of recommending one over another. While I did not list many objections, I did provide a proposition, which I believe reaches the heart of the matter. Various measures of asymptotic behaviour prevail (see Brent [1] and his references), but they do not remove the pathology. Moreover, as one abstracts the measurement of speed into more complex forms, one diminishes the chances of ever deducing the measure of a particular algorithm. Thus, I conclude that no simple, computable measure exists which can simultaneously satisfy the two axioms:

1. $\text{measure}(A) > \text{measure}(B)$ implies $A \succ B$
2. $\left\| A^k x - x^\infty \right\| < \left\| B^k x - x^\infty \right\|$ for all $k$ only if $A \succ B$.

My second point showed two areas where I believe rate analysis really has value: (1) acceleration (by replacing an application of $A$ with its asymptotic form), and (2) design of cut-off rules (where error bounds are important).
References


