本州大学紀要第2号（昭和48年3月）

近似値と誤差の考察

The Consideration of the Approximate Value and Errors

平井 由土
Yoshito Hirai

1 前書き

計算機を使用して，近似計算をする場合，その出力された値が，真値に対して，どれほどの誤差の範囲に存在するか等の，一連の計算機使用上において，若干観察したことにつき記述する。

いずれも，現状においては，未確定な課題を包括しており，これらのノートは，専門家諸氏の参考を考えており，かつこれらのものは，実務上応用し，検証されたい。情報工学の発達によって，数値解析学は，再検討された。そして，その本性ともいうべき誤差の問題は難関となっている。

ここで，実務家という語を用いるが，実務を離れての工学はありえないからである。

2 近似値の信頼性

ある近似値が与えられた場合，その真値に対する誤差を与えて示す方法は，きわめて抽象的な表現をまぬがれず，多くの実務家にとっては取り扱い難いものである。すなわち，よく見られるように，誤差 $\Delta x=10^{-n}$ 等の表示では，具体的取扱いに適せず，ないしは応用工学上不便である。

そこで，一般に x の小数第 n 位未満を，4捨5入して，近似値 a を得たとするとき，a は小数第 n 位まで正しいという表現を使い，またそれは丸めの誤差の限界として $|\Delta a|\leqq(1/2)\times10^{-n}$ が成立していることを前提とするものである。ただし $|x-a|=\Delta a$。

応用工学上，第 n 位まで正しいとは，このことである[1]。

ここに与えられた近似値の第 n 位まで正しいということの意義について若干述べる。これはこの注意書きの前提となるからである。

与えられた近似値が，一体，第何位まで，真値と一致しているか否かの問題は，この真値に対する近似値の誤差の限界をみることによってのみ知り得る。

たとえば，$x=1.248567\cdots\cdots$ を真値としたとき，$a=1.24854$ なる近似値は，誤差の限界が，$|\Delta a|\leqq10^{-4}/2$ で押えられるから，この近似値は小数第4位まで正しいというのである（田中明雄氏）。また $b=1.24860$ なる近似値についても，誤差の限界は等しく $|\Delta a|\leqq10^{-4}/2$ で押えられるがために，小数第4位まで正しいといえる。

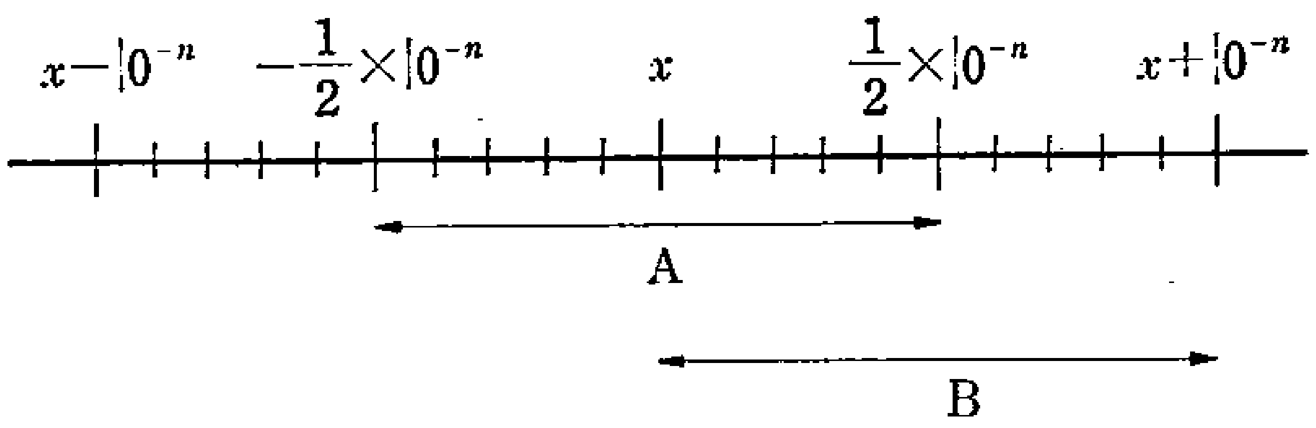
すなわち，数値解析上の小数第 n 位まで正しいということは，数値表現上の小数第 n 位まで数が一致するということではない。これは，非常に誤り易い所である。

数値表現上，正しいかどうかということよりは，真値といかなる距離のへだたりが存在するかが問題とされているといわねばならない。

一般に次のことが結論される。すなわち，誤差が $10^{-4}/2$ の範囲内で押えられる時，その近似値の真値に対する評価としては，第（$n-1$）位までは数値表現上一致するが，第 n 位は1つずれていることもあり，ただ真値に対して，$10^{-n}/2$ の

範囲において，ずれていることは，維持されている。

第1図　誤差評価の方法



今 a を真値として，A，B，2領域を考える。第1図において，Aの範囲に近似値が存在する時，その近似値は第 n 位まで正しいといい，Bの範囲に近似値が存在する時は，第 n 位まで真値と数値表現上一致しているというわけである。

いずれの場合も第（$n-1$）位においては，表現上も事実上も正しいわけである。ここで n は，整数 a は近似値で x は真値とする。

n は正整数としなかったのは，n が負でも，このことは成立する。すなわち，非常に大きい数の近似値問題についても，このことは成立する。

3　誤差評価方法のプログラムへの応用

一般に，プログラミングによって，なんらかの数値解析の計算を行なう場合，所詮，反復計算法が多く用いられる。この反復計算法は，つきつめて見ると収束数列の問題であるわけである（発散は除外してよい）。実務上の問題としては，収束しか考えられないからである。反復計算を行なってゆくとき，それは，

x_n，x_{n+1}，x_{n+2}，……

という収束数列を見るわけであるが，この場合，この出力されてきた値が，求めるべき真値に対して，いかなる誤差の範囲にあるか，一体，どこの値の何位までが正しい値であるのか，この問題が生ずるわけである。そこで，私は次のように考えたのである。

相対誤差は，

$\varDelta a=|x-a|$

とした時，

$|\varDelta a/x|$

ここで x は真値とする。すなわち，真値に対する絶対誤差の割合として表示される。しかし，実務上，真値はわからないのであり，既存の技術上知り得た数値をもとにして，誤差の限界を評価せねばならない。これまでの誤差を求める式は，

$$\varDelta p=\frac{x_n-x_{n+1}}{x_{n+1}}$$

等が利用された。x_n は計算機から出力された連続数列である。そして，$|\varDelta p|<10^{-m}$ 等を単に論じているにすぎない[2]。これは，実務家にとって抽象的でありすぎる。私が1．において述べたような，出力された近似値 x_n に対する信頼性，すなわち第何位まで正しいといえるか，何位まで真値と数値表現上一致しているかは理解しがたい。そこで，つぎのようにしてみる。x を真値として，x_n を第 n 回反復計算上出力されてきたものとする時，

$$\left|\frac{x_n-x_{n+1}}{x_{n+1}}\right|=\left|\frac{x_n-x+x-x_{n+1}}{x_{n+1}}\right|$$

$\lim_{m\to\&} x_n=x$ は，前提として成立しているものとするから，上式は，

$$\left|\frac{x_n-x_{n+1}}{x_{n+1}}\right|\leqq\left|\frac{x_n-x}{x_{n+1}}\right|+\left|\frac{x-x_{n+1}}{x_{n+1}}\right|$$

が成立している。∵ ε を任意の小さな数とする時，$|x-x_n|<\varepsilon$ ならしめる $n>N$ なる整数 N の存在。

$$\therefore\quad\left|\frac{x_n-x}{x_{n+1}}\right|\leqq\frac{1}{2x_{n+1}}\times10^{-m}$$

$$\left|\frac{x-x_{n+1}}{x_{n+1}}\right|<\frac{1}{2x_{n+1}}\times10^{-m}$$

を成立せしめる最大の m の値を見出す時，x_{n+1} は第 m 位まで正しいといえる。最大の m をさがす方法は，いかなる方法をとってもよい。

すなわち，

$$\left|\frac{x_n-x_{n+1}}{x_{n+1}}\right|<\frac{1}{x_{n+1}}\times10^{-m}\text{ —— （結論式）}$$

なる時，x_{n+1} は第 m 位まで正しいといえる。

なぜなら一般に，相対誤差の評価式は，

$$\left|\frac{x-a}{x}\right|\leqq\frac{1}{2a}\times10^{-m}$$

で示せるからである。

ここで a は近似値，x は真値である。

これが成立している時，近似値 a は第 m 位まで正しいといえるのである。

以上より，次の結論が導かれる。

計算機による反復計算の誤差評価方法は，

$$\frac{x_n - x_{n+1}}{x_{n+1}}$$

ないし，

$$\frac{x_n - x_{n+1}}{x_n}$$

を用いた時（いずれを用いても同じである），その値が $10^{-m}/x_n$ より小で押えられる時，x_n は第 m 位まで正しいと推定されるということである。

後に，簡単な例をかかげて，このことの成立を示した。

Newton-Raphson の平方根を求める方法において，この結論の実験を試行した。結果は，成立していることを示している。諸氏も，機会があったらいろいろな場合につき，試みられたい。

4 代入誤差の問題

あるプログラムを作成したとする。初期データや，投入データを入力して，一定の複雑な計算を行ない，結果を出力する。あらゆる形態の問題はこのパターンに合致する。しかし，投入データそのものが近似値の場合，もちろん計算結果も近似値であるにすぎない。一体，計算して出て来た結果の近似値の信頼性というものは，どのようになるか，これは特にシュミレーションを行なうときに悩む難問となる。計算機が倍精度を使ったからといって，解決するものではない。この点についても，若干の注意を明らかにしたい。この問題は，数値解析学上の代入誤差の問題そのものである（ただし，計算機のアキュムレータが発生させる誤差は別である）。

代入誤差の一般式は関数 $f(x_1, x_2, \cdots, x_n)$ に対し，

$$\Delta f = fx_1 \cdot \Delta a_1 + fx_2 \cdot \Delta a_2 + \cdots\cdots + fx_n \cdot \Delta a_n$$

したがって，相対誤差は，

$$\frac{\Delta f}{f} = \frac{fx_1}{f} \cdot \Delta a_1 + \frac{fx_2}{f} \cdot \Delta a_2 + \cdots\cdots + \frac{fx_n}{f} \cdot \Delta a_n$$

Δf は代入誤差であり，$\Delta a_i (i=1 \sim n)$ は絶対誤差（投入データ等がもっている誤差）。ただし，$fx_i (i=1 \sim n)$ は x_i に関する偏微分とする。

一般に，たとえば，小数第 n 位まで正しい近似値をもつものとして計算した結果は，必ず第 n 位以下（$n+1$ 位，$n+2$ 位……）は，正しくないとはいえないし，第 n 位以上（$n-1$ 位，$n-2$ 位……）も，正しいとも正しくないともいえない。正しいか正しくないかは，この偏微分係数を求め，Δf を求めずしては論ずることはできない。そこで，一つ具体例から簡単な事例をみてみる。

近似値 a_1，a_2，……，a_n を投入データとする。その誤差を Δa_1，Δa_2，……，Δa_n のとする。

$$|\Delta a_i| \leqq 10^{-m}/2 \quad (i=1 \sim n)$$

で得たとき，

$$\Delta f = \sum_{i=1}^{n} fx_i \times \Delta a_i$$

だから，

$$\Delta f \leqq \left(\sum_{i=1}^{n} fx_i\right) \cdot \frac{1}{2} \times 10^{-m}$$

したがって，

$$\left|\sum_{i=1}^{n} fx_i\right| = |fx_1 + fx_2 + \cdots\cdots + fx_n| \leqq 1 \qquad ①$$

の場合と，

$$\left|\sum_{i=1}^{n} fx_i\right| > 1 \qquad ②$$

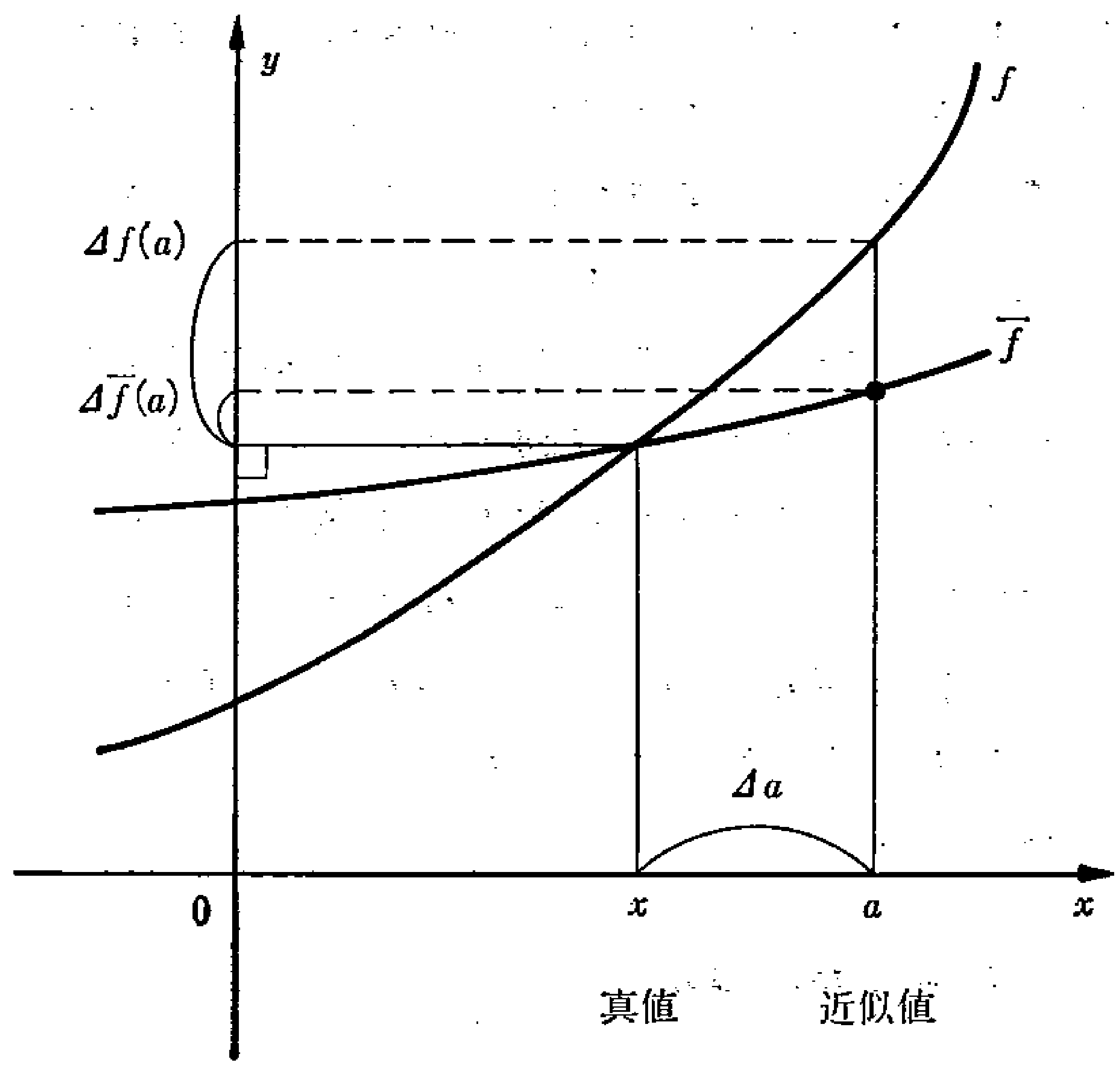
の場合によって Δf は大きく変化するわけで，この値によって評価は異なってくる。

①の場合は，$|\Delta f| \leqq 1/2 \times 10^{-m}$ が成立し，第 m 位まで得られた結果は信ずるに値する。

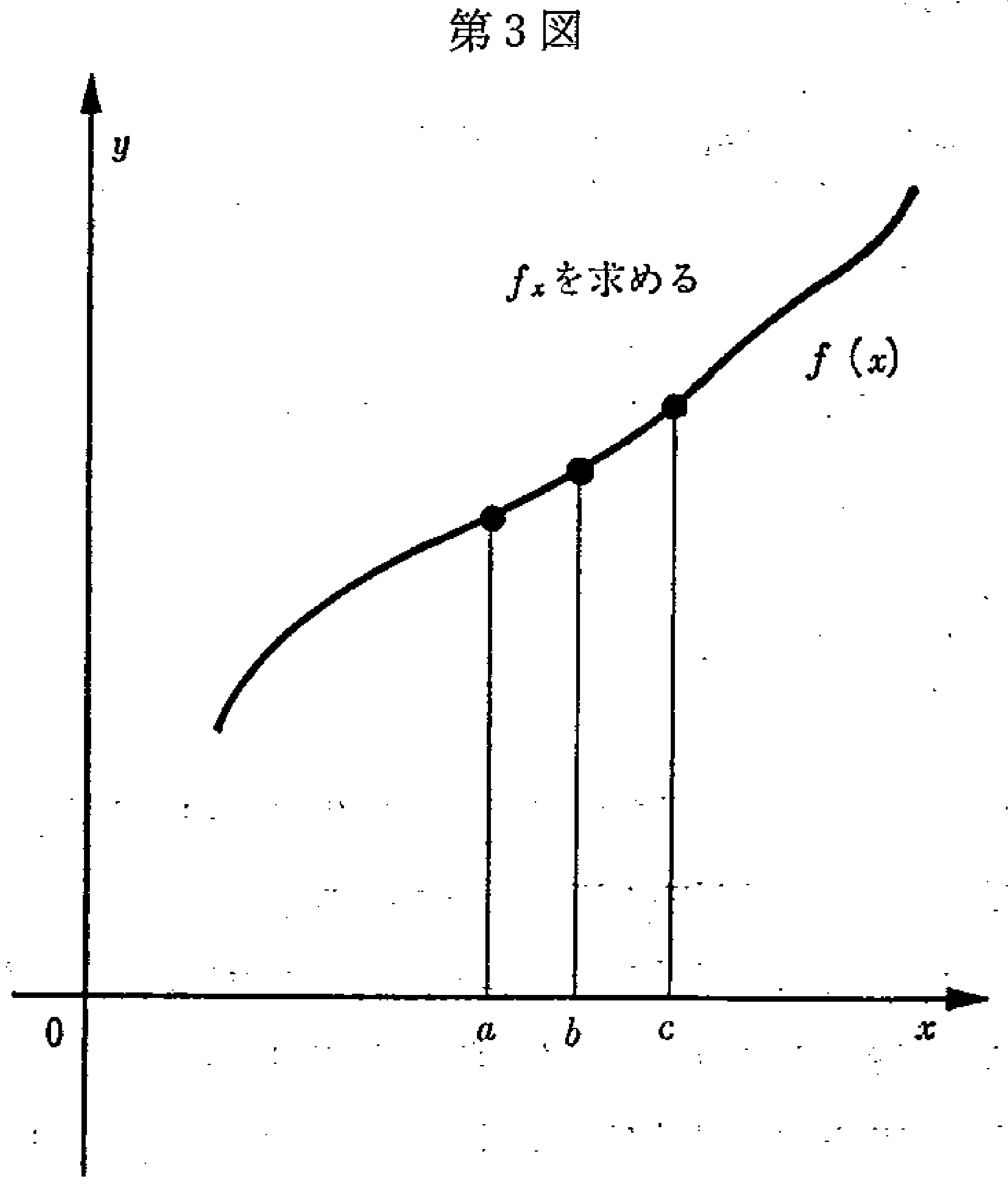
②の場合は，$|\Delta f| \geqq 1/2 \times 10^{-m}$ であり，これだけをもっては計算結果の値は確実な真値とは推定しがたい。すなわち，一概に断定することはできない。したがって，この $\sum_{i=1}^{n} fx_i$ の値の大きさがポイントとなる。この値の範囲，この値が，いいかえれば，どのような値でおさえるか，すなわち，最大値を求める問題と同値である。ここで，**OR** においてのすべての問題が最大値（裏がえして最小値）を求める問題であることを想起しなければならない。したがって，得られた値の信頼性の評価問題は，与えられた関数の存在範囲を求める問題へとつながっている。

実務上，与えられた数式の偏微分係数を求めることはきわめて困難である。そこで，近似値に近い値，b_i $(i=1 \sim n)$，c_i $(i=1 \sim n)$ 等を投入し，

第２図



第３図

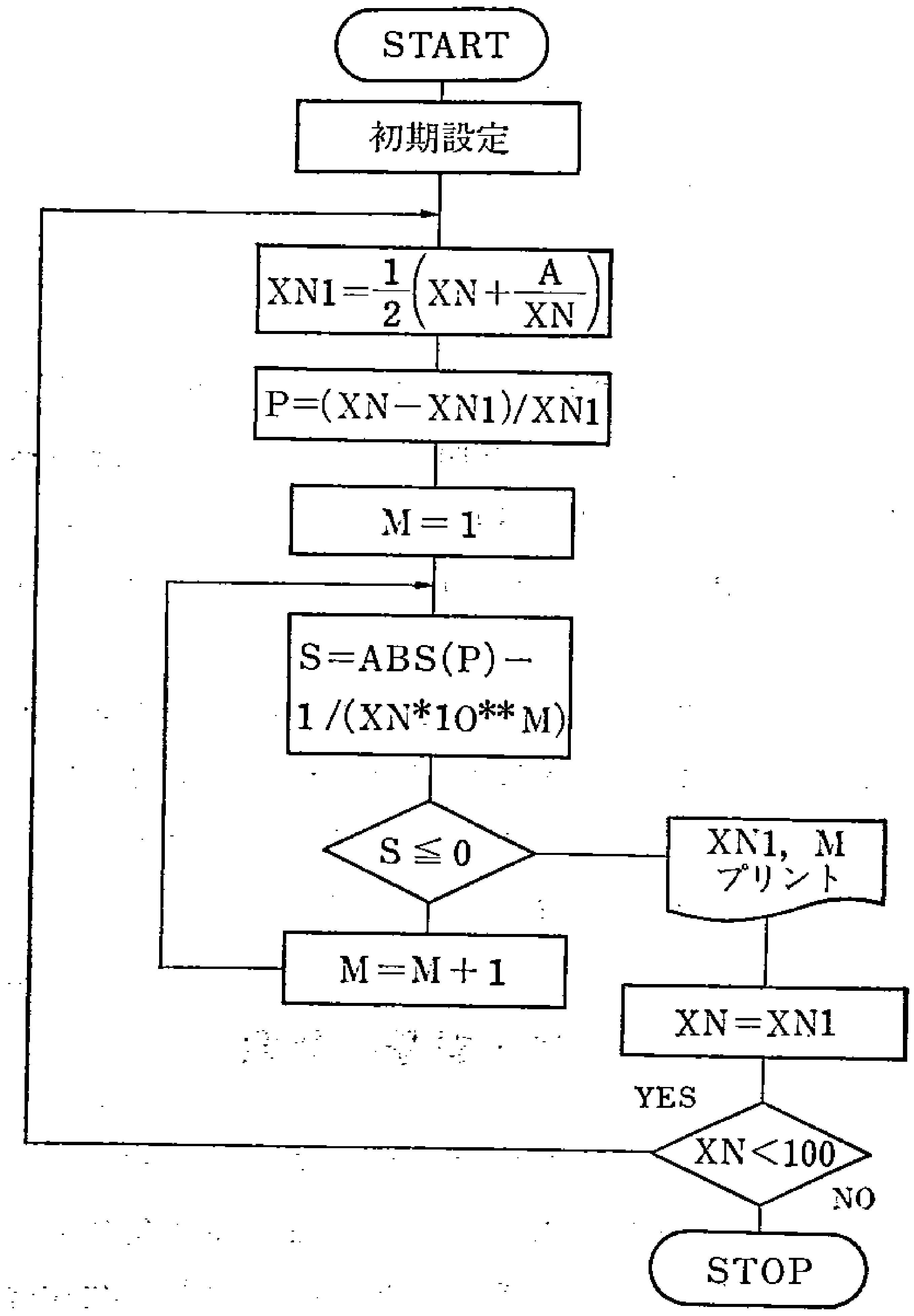


近似関数を描き，その偏微分値を，近似的に求める方法などが考えられる。

第２図においては，真値に対する誤差 $\varDelta a$ の大小と，計算後に発生する代入誤差との関係を示した図であって，微係数がいかに誤差に関係するかを示したものである。$\overline{f}$の関数のように微係数が小さいときは，$\varDelta a$ の値がいかに大きくとも，代入誤差はほとんど零に近いこともあり得るわけである。

ひるがえって思うに，変動現象下における誤差は，わずかな誤りでも大きな誤った結果を導く現

第４図　**プログラミング・フローチャート**



象はこの微係数による代入誤差の結論そのもののように考えられるのである。

一応，今回までの試みより，この誤差の問題の評価方法は，最大・最小を求める問題と密接に結びついているものといえる。

プログラム説明：Mの値は小数第M位までが，信頼できることを示した例である。出力されたデータの値の信頼性が，一見して解せるように工夫されたものである。

参　考　文　献

(1)　田中明雄：応用数学――数値計算法――p. 1～2，槇書店，1966. 3. 1.

(2)　Raymond W. Southworth and Samuel L. Delleuw : Digital Computation and Numerical Methods, p. 146～148. McGraw-Hill Book Company, New York, U.S.A., 1965.

```
FACOM 231   FORTRAN V-2    (SOURCE PROGRAM LIST PAGE 1)
*2804
        XN#2.0
        A#5.0
        DO 10 N#1,100
        XN 1# (XN+A/XN) /2.0
        P# (XN-XN 1) /XN 1
        M# 1
     15 S#ABSF (P) -1.0/ (XN*10.0**M)
        IF (S) 20, 20, 30
     20 M#M+1
        GO TO 15
     30 PRINT 100, XN1, M
    100 FORMAT (1H2, 5X, 3HXN#, F28, 25, 5X, 2HM#, 14)
        XN#XN 1
     10 CONTINUE
        STOP
        END
 XN#  2,2500000000000000000000000  M#   1
 XN#  2,2361111111111111111111111  M#   2
 XN#  2,2360679779158040027605245  M#   5
 XN#  2,2360679774997896964478728  M#  10
 XN#  2,2360679774997896964091737  M#  20
```