Title	経済分析におけるコンピューターの誤差問題(その1) : 誤差問題の重大性とその分析方法
Sub Title	Errors generated by computer in economic analysis
Author	鳥居, 泰彦 新井, 益洋
Publisher	慶應義塾経済学会
Publication year	1975
Jtitle	三田学会雑誌 (Keio journal of economics). Vol.68, No.6 (1975. 6) ,p.553(53)- 562(62)
JaLC DOI	10.14991/001.19750601-0053
Abstract	
Notes	研究ノート
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00234610-19750601-0053

経済分析におけるコンピューターの誤差問題 (その1)

——誤差問題の重大性とその分析方法——

鳥居 泰彦
新井 益洋

1. 問題提起

私達2人の著者は，実証的な経済分析の仕事に携っている間に，コンピューターの精度問題が分析の精度に非常に大きな（というよりはむしろ決定的な）影響を及ぼすことを痛感した。それは，時には，経済理論，エコノメトリクス，統計理論等の理論的精度をはるかに超える大きさで私達の分析結果を攪乱することさえある。私達2人の著者は，以前にも一度この問題をとり上げて注意を喚起したことがあるが，ここに同じ問題を再提起して読者の注意を促したい。

1960年以後の経済分析の進歩は実に著しい。第1に，経済理論そのものが精密化された。第2に，実証的な経済分析の方法論であるエコノメトリクスが進歩した。第3に，高速・大容量のコンピューターが普及した。これら3つの進歩によって，大量のデータを処理し経済理論の仮説を確めることが可能になった。また，大型の複雑な経済モデルを測定して実用的な経済モデルを作成し，実際の政策判断に用いることが可能となった。

既に，多くの国の政策当局が，経済成長や景気変動の過程を解明し，経済の運行をシミュレートするためのエコノメトリック・モデルを作成している。これは，明らかに，ケインズ理論とその後の新古典派理論の発展，エコノメトリクスの進歩，コンピューターの発達と普及によるものである。また，レオンティエフが開発した産業連関分析も，厖大な基礎統計の収集・集計と大規模な行列演算がコンピューターの力を借りて可能となった為に，今では多くの国の政策当局が実際に採用するようになった。このように，経済分析はもはや机上の理論ではなく，現実の経済問題の解決のために本格的に応用される段階に至っている。

今後は，私達が直面する現実の経済問題は多分，今迄より一層大掛りな経済分析を必要とするようになるであろう。たとえば，国際通貨・金融問題，開発途上国援助と産業の国際分業等の問題は，各国の産業連関表と世界貿易マトリクスの連動といった途方もなく大規模な分析を必要としている。また，今後に予想される各国の現実的な経済計画の方向は，複雑なポリシー・ミックスと産業構造政策の選択に深く立ち入る必要に直面している。これらはいずれも，今迄とは比較にならないほど高度で大規模な分析計算の時代が到来することを示唆している。

ここに困った問題がある。コンピューターを使いさえすれば経済理論とエコノメトリクスが教えている通りの分析計算が実行できるかというと，実はそうではない。実はコンピューターは私達が考えているほど正

(注) この研究ノートは，慶應義塾大学産業研究所および情報科学研究所において鳥居・新井が行なっている共同研究の成果の一部である。

この研究ノートは鳥居・新井として既に発表したものに加筆訂正をした。この問題について多くの人々の注意を喚起するためにあえて再論することとした。

この，コンピューターの精度問題の分析には，コンピューターの有効桁数を色々に変えて精度実験を行なう必要があった。この特殊な実験の為に，世界でおそらくただ1台になってしまった古典的なコンピューターである慶應義塾大学情報科学研究所のIBM 1620が，「variable length machine」としての機能を発揮した。この古い機械の維持の為に研究所とIBM社が払ってきた努力に感謝したい。

確無比な計算用具ではない。使い方によっては，1+1の答が2にならないことさえある厄介な計算用具である。今までに随分沢山のエコノメトリクスの解説と研究の書物が刊行されたし，随分沢山の実証分析が行なわれたが，これらは，ごく少数の例外を除いて，コンピューターさえ使えば完全に理論通りの計算ができるという盲信の下に行なわれていた。

私達2人の著者は，コンピューターを使って経済の分析計算をする場合，計算誤差がいかに大きな問題を提起するかを示そうと思う。そして，計算誤差の大きさを探る方法を示そうと思う。その上で，経済分析に最も頻繁に用いられる分析計算の1つである回帰分析を中心に，コンピューターの誤差問題が提起する困難をどのように処理するかを考えてみたいと思う。

2. コンピューターから発生する計算誤差の実例

コンピューターが「計算間違い」を犯すことは殆んどない。しかし，その答は，「ほとんど常に誤差を含んでいる。」計算を間違いなく実行するということと計算に誤差が含まれることとは別のことである。このことは，コンピューター・サイエンティストにとってはごく当り前のことであるために，コンピューターの専門家でないユーザーの目に触れる形で議論が行なわれなかった。経済研究者の間でも，R. Hall〔2〕やLongley〔1〕等のごく少数の例外を除いて，一般にはコンピューター誤差問題は省みられなかった。

以下3のつの具体例によって，コンピューターとはかくも油断のできない代物であるということを認識して頂きたい。

〔具体例1〕

2次方程式の根を求める問題

$$x^2+0.4002x+0.00008=0 \qquad (2.1)$$

簡単な問題である。2次方程式の根の公式

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a} \qquad (2.2)$$

に(2.1)式の係数（$a=1$, $b=0.4002$, $c=0.00008$）を代入すれば根が計算できる筈である。

いま試みに(2.2)式の大きい方の根$\frac{-b+\sqrt{b^2-4ac}}{2a}$を手計算で正確に計算してみよう。正確な答を知るために，4捨5入，切り上げ，切り捨て等のまるめの操作（rounding）を行なわないで計算すると，答は下のようになる。

$$x=\frac{-0.4002+\sqrt{0.4002^2-4\times0.00008}}{2}$$

$$=\frac{-0.4002+\sqrt{0.16016004-0.00032}}{2}$$

$$=\frac{-0.4002+0.3998}{2}$$

$$=-0.0002$$

この根（$x=-0.0002$）を(2.1)式の左辺に代入すると左辺は正確に0となって右辺の0と等しくなるから，この根は正解である。

次に，同じ計算を，有効桁数が4桁のコンピューターで実行してみる。答は-0.00025となる。この答を(2.1)式の左辺に代入しても左辺は0にならない。この答は0.00005の誤差を持っている。正解（$x=-0.0002$）に対して25%の誤差率である。

コンピューターの有効桁数を4桁から5桁，6桁……と増やしてみると，同じ(2.2)式の計算結果は〔Table 1〕のようになる。〔Table 1〕には記入してないが，コンピューターの有効桁数を9桁，10桁……と更に上げて行っても，もはや誤差は生じない。

〔Table 1〕コンピューターによる計算結果

コンピュータの有効桁数	計算結果	誤差	誤差率
4 桁	−0.00025	+0.00005	25%
5 桁	−0.000205	+0.000005	2.5%
6 桁	−0.0002005	+0.0000005	0.25%
7 桁	−0.00020005	+0.00000005	0.025%
8 桁	−0.0002	0	0%

以上の実験から，(1)コンピューターの有効桁数がある程度以下であると計算誤差が発生する。(2)有効桁数を増やすにつれて誤差は小さくなる。(3)有効桁数がある程度以上になると誤差は発生しない。この実例から，計算誤差の発生の原因はコンピューターの有効桁数が不足していることに関係があるらしいということが推察されるであろう。

〔具体例2〕

マクローリン展開による$\sin\pi$の計算

$$\sin x=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\frac{x^9}{9!}-\cdots\cdots \qquad (2.3)$$

(2.3)式が有限のxについて$\sin x$の近似値を与えることはよく知られている。また，この公式は，「演算を途中で打ち切ることによって生ずる真の値と近似値の誤差の絶対値は切捨てた最初の項の絶対値よりも小さい」ことが証明されている。

ところで，私達は $\sin\pi=0$ であることを予め知っている。コンピューターの精度がどの程度のものであるかを確めるために計算をさせた結果を〔Table 2〕に示してある。有効桁数が16桁のコンピューターでも $\sin\pi=0$ の正解にくらべてかなり精度が悪いことがわかる。

この実験から，誤差の理由には(1)コンピューターの有効桁数の不足だけでなく，(2)演算の中止も関係しているという推察ができる。

〔Table 2〕 $\sin\pi$ の計算

コンピューターの有効桁数	$\sin\pi$ の計算結果
4 桁	0.001033
6 桁	0.000014099
8 桁	0.00000009465……
16 桁	0.00000001999……

〔具体例3〕

2元連立方程式の解を求める問題

$$\left.\begin{aligned}5x-331y&=3.5\\6x-397y&=5.2\end{aligned}\right\}\qquad(2.4)$$

この問題の解を手計算で正確に求めた結果は

$$\left.\begin{aligned}x&=331.7\\y&=5.000\end{aligned}\right\}\qquad(2.5)$$

である。この解を (2.4) 式に代入してみれば正確に左辺と右辺が等しくなって，正しい解であることが確められる。

この問題について興味深い実験を行なおう。(2.4) 式の下の式の右辺に観測誤差があって5.2が5.1となってしまった場合を想定する。誤差率にして約2%の観測誤差である。このためにコンピューターに解かせる連立方程式は (2.4) 式の代りに (2.6) 式となる。

$$\left.\begin{aligned}5x-331y&=3.5\\6x-397y&=5.1\end{aligned}\right\}\qquad(2.6)$$

この(2.6)式の解は

$$\left.\begin{aligned}x&=298.6\\y&=4.5\end{aligned}\right\}\qquad(2.7)$$

となって，もとの解 (2.5) 式とは大分かけ離れたものとなる。つまり，元の方程式の係数の1つにわずか2%の観測誤差があったために，解は10%近い誤差を含んでしまったことになる。

次に，同じ (2.4) 式についてもう1つの実験をしてみる。コンピューターの有効桁数を5桁にしておいて (2.4) 式に次の値を代入する。

$$\left.\begin{aligned}x&=358.18\\y&=5.4\end{aligned}\right\}\qquad(2.8)$$

(2.8)式は正解(2.5)式にくらべればあまりにかけ離れていることは一目瞭然であるが，これを (2.4) 式に代入すると，実に驚くべきことだが左辺＝右辺が成立してしまう。コンピューターの有効桁数が変ると，全く違う数値が正解となり得るのである。

この例題の (2.4) 式は，グラフに描いてみると非常に接近した2本の方程式の連立解を求める問題であることがわかる。このように，問題の性質からくる誤差もあることに注目する必要がある。

3. 誤差の源泉

以上の具体例と実験を通じてみると，次の4つのタイプの誤差が発生していたことがわかる。

(1) コンピューターの有効桁数の不足による誤差
(2) 演算打切りによる誤差
(3) 数値（データ）に固有の誤差とその増幅による誤差
(4) 問題の性質に由来する誤差

以下では，これらの誤差についてみることにするが，それに先立って誤差を次のように定義する約束とする。

絶対誤差

$$\varepsilon_{\hat{X}}=X-\hat{X}$$

相対誤差

$$\frac{\varepsilon_{\hat{X}}}{\hat{X}}=\frac{X-\hat{X}}{\hat{X}}$$

但し，X は真の値，$\hat{X}$は誤差を含む計算値である。なお，相対誤差の計算において，一般には，分母は$\hat{X}$ではなくXを用いる。しかし今，私達にとって真の値Xは未知数である。そこで，相対誤差の定義には分母にX（真の値）の代りに$\hat{X}$(計算値) を用いることにする。

3.1 Rounding Error

コンピューター誤差の中で最も重要な意味を持つのは rouning error（まるめの誤算）である。その他の種類の誤差（後述）が全くない場合でも rounding error だけは避けられない。

rounding error とは何かを知るためには，コンピューター内部で数値がどのように記憶されるかを知っておく必要がある。コンピューターはどのような数値でも扱えなければならない。正の数値と負の数値は勿論

のこと，絶対値の大きい数も小さい数も，桁数の多い数も少ない数も同一の基準で扱うことができなければならない。

商業ベースの製造コストの範囲内でこの目的を果たすために，一般のコンピューターは数値の基準化または正規化（normalization）を行なって，コンピューター内部での数値表現方法を統一している。いま理解しやすいように10進法のコンピューターを仮想して説明すると，数値の正規化というのは，「どんな数値でも小数第1位から始まる数値と10の冪乗との積として表現すること」である。たとえば

13570000.0

という数値は

$.1357\times10^{8}$

として表現する。同様にして

−0.00000013579

という数値は

$-.13579\times10^{-6}$

として表現する。即ち，いかなる数値も有効数字の部分と小数点位置を調整するための指数部分の組み合わせで表わすのである。有効数字を表わす部分を mantissa（仮数部），10 の何乗倍であるかを表わす部分を exponent（指数部）と呼ぶ。

ところで，正規化には不可避的に rounding error がつきまとう。それは，仮数を一定の桁数に制限しなければならないからである。特殊なコンピューターは別として，一般に用いられている汎用コンピューターでは mantissa は（10進法に換算して）7桁～18桁である。

たとえば 123456543210.0 という12桁の数値を mantissa が8桁のコンピューターに格納すれば

$.12345654\times10^{12}$

として格納することになるから，右端の 3210.0 は切り捨てられてしまう。これが rounding error である。

実際の汎用コンピューターの大部分は10進法ではなく，2進法，8進法，16進法等の方法で数値表現をしているので，10進法の場合よりもやっかいな問題が起り易い。たとえば，10進法の 1/10 は 0.1 であるから mantissa は1桁あれば充分に格納できるが，同じ 1/10 を2進法で表現すると

$0.0001100110011\cdots\cdots_{(2)}$

という無限循環小数になるから，2進法のコンピューターでは 1/10 の記憶に rounding error がつきまとうことになる。このような例は枚挙にいとまがない。

rounding error は，単に数値の格納だけでなく，2つ以上の数値の加減乗除を行なうと更に深刻な問題になる。即ち，演算実行前には mantissa の範囲内に収まっていた数値も演算の結果前より大きな有効桁数を持つようになるから，演算結果を格納するためにはどうしても rounding を行なわなければならないことになる。rounding の操作としては，「切り捨て」，「切り上げ」，「四捨五入」等が考えられるが，市販されている汎用コンピューターは殆んどが「切り捨て」方式を採用している。

4章では，この「切り捨て」によって生ずる誤差の大きさを評価する方法を提案する。

3.2 Trancation Error

前述の〔具体例2〕でみたように，いつまでも級数計算を続ければ正確な答が得られることが判っていても，実際の演算はどこかで打ち切らなければならない。演算の打ち切り（trancation）は，やはり不可避的に計算結果の誤差を生む。

最近の経済分析計算は，非線型モデルを始めとして，繰り返し計算（iteration）を必要とするものが増えている。これらの iteration を含む計算では，問題ごとに trancation error の大きさを評価しながら演算打切りの限度をみきわめなければならない。

もう一つの問題がある。FORTRAN に代表される各種コンパイラーには，log, exp, tan 等のライブラリー・サブルーチンが組み込んである。これらのサブルーチンの多くは，級数展開によって計算を行なっている。そのために前述の trancation error の不安が残るのである。

例えば，Cobb-Douglas 型の生産関数

$$Q=aK^{\alpha}L^{\beta} \qquad (3.1)$$

を計測する場合を考えてみると，普通は

$$\log Q=\log a+\alpha\log K+\beta\log L \qquad (3.2)$$

のように対数データについての線型回帰分析を行なう。データの数の3倍（Q,K,L）だけの対数計算（級数展開）をするのであるから，そのたびに trancation error が発生していることになる。この場合には，データの段階からして既に trancation error の介入が相当にあることを予想しなければならない。

3.3 Inherent Error

数値（データ）自身がさまざまな誤差を持ち得る。経済統計の場合には，データに固有の誤差は無視できない。inherent error は主として次の3つの理由で発生

する。

第1に，観測誤差または測定誤差である。経済統計の場合には，自然科学の観測誤差にくらべて観測誤差の可能性は大きい。

国勢調査や事業所センサスは悉皆調査であるが，それでも「調査漏れ」，「二重記載」，「誤記入」等の誤差が起る。まして大部分の経済統計は悉皆調査ではなく**標本調査**であるから，抽出した標本が正確に母集団を代表していないために生ずるところの「標本抽出誤差」を含む。

第2に，統計データの取り扱いのミスによって生ずる誤差がある。調査原票の記入ミス，統計表のミスプリント，データの転記ミス，カードパンチのミス等，主として不注意による誤りである。

不注意によって生ずるミスは，原理的には注意深く作業をすることによって防ぐことができるが，不注意による誤りを減らそうとすれば作業速度が低下する。そこで，ある程度の不注意ミスは止むを得ないものとして容認してしまって，ミスの発見と事後処理の合理的な作業システムを考える方がよい。このことは，経済分析のためのデータ・ベースの設計と管理を考える上で大切な点である。

第3に，統計数値を記述する際の限界の問題がある。たとえば，仮りに日本の人口を最後の桁まで正確に把えることができたとしても，それを台帳に書き留めたり，印刷したり，カードにパンチしたり，磁気テープに記録したりする場合に，完全に最後の桁まで記録し切れない。実際には，千人とか万人といった単位で表示してそれ以下の桁を捨象することが多い。このように，表現可能な桁数に限度があるために，データが固有の誤差を持ってしまうのが実情である。

4. Rounding Error の性質

4.1. Rounding Error の定式化

コンピューターが不可避的に rounding error を発生する理由は3.1節で述べた通りである。rounding error の大きさは全く at random に決定するものであって，そこには何の規則性も見出せないように考えられがちであるが，よく調べてみると，そこにはある種の規則性が存在する。

rounding error の性質を定式化するために，次の記号を用いることにする。

X： 真の値

m_X： 仮数部の値

e： 指数部の値

E_X： 誤差を正規化して表わした仮数

t： 仮数の桁数

説明をわかり易くするために，しばらくの間10進法のコンピューターを想定する。3.1 節で説明した正規化の方法によって考えれば，仮数部 (mantissa) m_X は次の値をとる筈である。

$$\frac{1}{10}\leqq|m_X|<1 \qquad (|X|\neq 0 \text{ の時}) \tag{4.1}$$

$$m_X=0 \qquad (|X|=0 \text{ の時}) \tag{4.2}$$

E_X も一種の仮数であるが誤差が0の場合もあり得るから

$$0\leqq|E_X|<1 \tag{4.3}$$

以上の記号を使って真の値を表わせば

$$X=m_X\cdot 10^e+E_X\cdot 10^{e-t} \tag{4.4}$$

と表わすことができる。

(4.4)式の右辺第2項の冪 $e-t$ は rounding によって発生した誤差をもう一度改めて正規化するために小数点を t 桁だけ左に移動させたことを示している。

(例)

mantissa が4桁のコンピューターに6桁の数値 12345.6 を格納するとすれば

$m_X=.1234$

$e=5$

$E_X=.56$

$e-t=5-4=1$

誤差 $E_X\cdot 10^{e-t}$ は「切り捨て」，「切り上げ」，「四捨五入」，「偶捨奇入」等の rounding によって発生する。「切り捨て」や「切り上げ」にくらべて「四捨五入」や「偶捨奇入」の方が望ましいことは後述する。

4.2. 切り捨て誤差の性質

はじめに，切り捨て誤差の最大限界を評価できることを示そう。4.1節で約束した記号に加えて次の3つの記号を約束する。

$\hat{X}=m_X\cdot 10^e$ ： 切り捨てを行なった数値

$\varepsilon_X=E_X\cdot 10^{e-t}$ ： 絶対誤差

$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 10^{e-t}}{m_X\cdot 10^e}\right|$ ： 相対誤差

(4.1), (4.2), (4.3)式で示したように，E_X と m_X はそれぞれ $0\leqq|E_X|<1$, $\frac{1}{10}\leqq|m_X|<1$ という範囲を持っているから，これを使って相対誤差の最大限を知ることができる。

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 10^{e-t}}{m_X\cdot 10^e}\right|\leqq\left|\frac{1\times 10^{e-t}}{0.1\times 10^e}\right|=10^{-t+1}=\frac{1}{10^{t-1}} \tag{4.5}$$

(4.5) 式は，切り捨ての相対誤差が次のような性質を持っていることを示唆している。

(1) mantissa が t 桁のコンピューターでは切り捨ての相対誤差は $\frac{1}{10^{t-1}}$ を超えることはない。

(2) 切り捨ての相対誤差の最大限は，コンピューターの有効桁数 (mantissa) が大きくなるにつれて小さくなる。

つまり，有効桁数が4桁のコンピューターでは，切り捨ての相対誤差の最大限は $\frac{1}{10^{4-1}}=0.1\%$ であるが，有効桁数が8桁のコンピューターでは $\frac{1}{10^{8-1}}=0.00001\%$ である。

このことは，コンピューターの有効桁数は事情が許す限り大きい方がよいことを示唆している。

4.3. n 進法コンピューターの切捨て誤差限界

前節では10進法のコンピューターを想定したが，実際に使われているコンピューターは2進法(binary)，8進法 (octal)，16進法 (hexadecimal) 等の数値表現を用いている。本節では2進法あるいは16進法の場合の切り捨て誤差の限界について述べておく。

2進法の正規化の操作は，次のように表わすことができる。

$$X=m_X\cdot 2^e+E_X\cdot 2^{e-t} \tag{4.6}$$

但し，ここでは，m_X と E_X はそれぞれ次の範囲を持った2進数である。

$$\frac{1}{2}\leqq|m_X|<1 \qquad (1\times 1\neq 0\text{の時}) \tag{4.7}$$

$$0\leqq|E_X|<1 \tag{4.8}$$

従って，2進法で t 桁の mantissa を持ったコンピューターで切り捨てを行なう場合の相対誤差の限界は，次のようになる。

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 2^{e-t}}{m_X\cdot 2^e}\right|\leqq\left|\frac{1\times 2^{e-t}}{\frac{1}{2}\times 2^e}\right|=2^{-t+1}=\frac{1}{2^{t-1}} \tag{4.9}$$

2進数の切り捨ての相対誤差の限界は，2進法で t 桁の有効桁数を持ったコンピューターにおいては $\frac{1}{2^{t-1}}$ である。この場合にも，コンピューターの有効桁数を大きくすれば相対誤差の限界は縮小する。

同じ方法で16進数のコンピューターの切り捨て誤差の限界を示すと次のようになる。

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 16^{e-t}}{m_X\cdot 16^e}\right|\leqq\left|\frac{1\times 16^{e-t}}{\frac{1}{16}\times 16^e}\right|=16^{-t+1}=\frac{1}{16^{t-1}} \tag{4.10}$$

同様にして，一般に n 進法のコンピューターの切り捨ての相対誤差の限界は

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|\leqq n^{-t+1} \tag{4.11}$$

である。但し，t は n 進法で t 桁の有効桁数に対応している。

4.4. 四捨五入の誤差の性質

前述したように，実際に用いられている汎用コンピューターは，ほとんど例外なく「切り捨て」による rounding を行なう。しかし「切り捨て」の代りに「四捨五入」を採用すると誤差の限界は半分になる。それは，次のようにして認めることができる。

「四捨五入」によって rounding した値を $\hat{X}$ とすると $|\hat{X}|$ は次のように定式化できる。

$$|\hat{X}|=|m_X|\times 10^e \qquad (|E_X|<0.5\text{の時}) \tag{4.12}$$

$$|\hat{X}|=|m_X|\times 10^e+10^{e-t} \qquad (|E_X|\geqq 0.5\text{の時}) \tag{4.13}$$

つまり (4.12) 式は「四捨」の場合を示し，(4.13) 式は「五入」の場合を示している。四捨五入の絶対誤差は次のように定式化できる。

$$|\varepsilon_X|=|E_X|\times 10^{e-t} \qquad (\text{「四捨」の場合}) \tag{4.14}$$

$$|\varepsilon_X|=|1-E_X|\times 10^{e-t} \qquad (\text{「五入」の場合}) \tag{4.15}$$

絶対誤差の mantissa E_X 範囲は，「四捨」の場合には $|E_X|<0.5$ であるから

$$|\varepsilon_X|<0.5\times 10^{e-t} \qquad (\text{「四捨」の場合}) \tag{4.16}$$

が成立つ。「五入」の場合には $|1-E_X|\leqq 0.5$ であるからやはり

$$|\varepsilon_X|\leqq 0.5\times 10^{e-t} \qquad (\text{「五入」の場合}) \tag{4.17}$$

が成立つ。従って，「四捨」，「五入」いずれの場合にも相対誤差の範囲は次のように定式化できる。

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|\leqq\left|\frac{E_X\times 10^{e-t}}{m_X\times 10^e}\right|\leqq\left|\frac{0.5\times 10^{e-t}}{0.1\times 10^e}\right|=0.5\times 10^{-t+1}$$

$$=0.5\times\frac{1}{10^{t-1}} \tag{4.18}$$

(4.18) 式と (4.5) 式とを比較すればすぐわかるように，四捨五入の相対誤差の限界は切り捨ての相対誤差の限界にくらべて半分になっている。勿論，四捨五入

の相対誤差もコンピューターの有効桁数を大きくすれば急速に縮小する性質があることには変りない。

結論的に，コンピューターの有効桁数は大きいほど rounding error を小さくすることができる。しかも，同じ rounding でも，一般に行なわれている「切り捨て」よりは「四捨五入」を採用した方が誤差の限界は半分になる。

なお，2進法の演算では10進法の「四捨五入」に相当する symmetric rounding は「0捨1入」である。この場合の相対誤差の範囲は次のようになる。

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 2^{e-t}}{m_X\cdot 2^e}\right|\leqq\left|\frac{\frac{1}{2}\cdot 2^{e-t}}{\frac{1}{2}\cdot 2^e}\right|=2^{-t} \qquad (4.19)$$

(4.9) 式の2進法の切り捨て誤差の範囲と比較するとちょうど半分になっていることがわかる。同様に16進法について調べてみると「8捨9入」の相対誤差の範囲は

$$\left|\frac{\varepsilon_X}{\hat{X}}\right|=\left|\frac{E_X\cdot 16^{e-t}}{m_X\cdot 16^e}\right|\leqq\left|\frac{\frac{1}{2}\cdot 16^{e-t}}{\frac{1}{16}\cdot 16^e}\right|=8\times 16^{-t} \qquad (4.20)$$

となって，前述の (4.10) 式のちょうど半分になっている。

5. 四則演算に伴う誤差の評価.

5.1. 記号

第4章では一つの数値を単独に rounding する場合の誤差の性質をみたが，コンピューターによる分析計算は非常に多くの四則演算の繰り返しである。計算ステップを重ねる度に誤差が複雑に組み合わさって行く。

多数の計算ステップからなる一連の計算をコンピューターを用いて行なうためには，誤差の累積についての何等かの規則性がわかっていなければならない。後述のプロセス・グラフは，誤差の累積状況をトレースするのに有効な方法であると考えられる。プロセス・グラフによる誤差累積の追跡法を理解する為に，四則演算によって誤差がどのように合成されるかを定式化しておこう。

以下の説明には，次の記号を用いる。

X, Y：真の値

$\hat{X}$, $\hat{Y}$：計算値

E_X, E_Y：X, Yの絶対誤差

E_{X+Y} ：和（X＋Y）の絶対誤差

E_{X-Y} ：差（X－Y）の絶対誤差

$E_{X\cdot Y}$ ：積（X・Y）の絶対誤差

$E_{X/Y}$ ：商（X／Y）の絶対誤差

5.2. 加算による誤差の合成

誤差を伴う2つの数値の加算は次のように定式化できる。

$$X+Y=(\hat{X}+E_X)+(\hat{Y}+E_Y)$$
$$=(\hat{X}+\hat{Y})+(E_X+E_Y) \qquad (5.1)$$

従って，和の絶対誤差 (E_{X+Y}) は次のように書ける。

$$E_{X+Y}=E_X+E_Y \qquad (5.2)$$

また，和の相対誤差は，次のようにして，個別相対誤差の加重平均であることを導くことができる。

$$\frac{E_{X+Y}}{\hat{X}+\hat{Y}}=\frac{E_X+E_Y}{\hat{X}+\hat{Y}}=\frac{\hat{X}}{\hat{X}+\hat{Y}}\left(\frac{E_Y}{\hat{X}}\right)+\frac{\hat{Y}}{\hat{X}+\hat{Y}}\left(\frac{E_Y}{\hat{Y}}\right) \qquad (5.3)$$

E_X, E_Y の符号は正負いずれもあり得るから，(5.2), (5.3) 式の合成誤差の大きさは様々であろう。E_X と E_Y の符号が同符号の時は E_{X+Y} は大きいし，異符号の場合には E_{X+Y} は0となることもあり得る。

5.3. 減算による誤差の合成

前節と同様にして，差の絶対誤差は個別誤差の差であることがわかる。

$$E_{X-Y}=E_X-E_Y \qquad (5.4)$$

差の相対誤差は個別相対誤差の加重平均である。

$$\frac{E_{X-Y}}{\hat{X}-\hat{Y}}=\frac{\hat{X}}{\hat{X}-\hat{Y}}\left(\frac{E_X}{\hat{X}}\right)-\frac{\hat{Y}}{\hat{X}-\hat{Y}}\left(\frac{E_Y}{\hat{Y}}\right) \qquad (5.5)$$

5.4. 乗算による誤差の合成

2つの数値XとYの積は次のように定式化できる。

$$X\cdot Y=(\hat{X}+E_X)\cdot(\hat{Y}+E_Y)$$
$$=\hat{X}\hat{Y}+\hat{X}E_X+\hat{Y}E_Y+E_X\cdot E_Y \qquad (5.6)$$

(5.6) 式の最終項の $E_X\cdot E_Y$ は mantissa の右端で発生した誤差の積であるから，積 $X\cdot Y$ にくらべると非常に小さい筈である。故に $E_X\cdot E_Y$ の項を捨象して(5.6)式を書き直すと，

$$X\cdot Y\simeq\hat{X}\cdot\hat{Y}+\hat{X}E_X+\hat{Y}E_Y \qquad (5.7)$$

従って積の絶対誤差は

$$E_{X\cdot Y}\simeq\hat{X}E_X+\hat{Y}E_Y \qquad (5.8)$$

また，積の相対誤差は

$$\frac{E_{X\cdot Y}}{\hat{X}\cdot\hat{Y}}\simeq\frac{\hat{X}E_Y+\hat{Y}E_X}{\hat{X}\cdot\hat{Y}}=\frac{E_X}{\hat{X}}+\frac{E_Y}{\hat{Y}} \qquad (5.9)$$

すなわち，$E_X\cdot E_Y$ が充分に小さければ（実際に充分に小

さい)，積の相対誤差は個別相対誤差の和に等しい。

5.5. 除算による誤差の合成

誤差を含む2数の間の除算は次のように定式化できる。

$$\frac{X}{Y}=\frac{(\hat{X}+E_X)}{(\hat{Y}+E_Y)}=\left(\frac{\hat{X}+E_Y}{\hat{Y}}\right)\left(\frac{1}{1+E_Y/\hat{Y}}\right) \quad (5.10)$$

(5.10) 式の右辺第2項は次のように級数展開できる。

$$\frac{1}{1+E_Y/\hat{Y}}=1-\frac{E_Y}{\hat{Y}}+\left(\frac{E_Y}{\hat{Y}}\right)^2-\left(\frac{E_Y}{\hat{Y}}\right)^3+\cdots\cdots \quad (5.11)$$

(5.11) 式を (5.10) 式に代入すれば

$$\frac{X}{Y}=\frac{\hat{X}+E_X}{\hat{Y}}\left\{1-\frac{E_Y}{\hat{Y}}+\left(\frac{E_Y}{\hat{Y}}\right)^2-\left(\frac{E_Y}{\hat{Y}}\right)^3+\cdots\cdots\right\} \quad (5.12)$$

(5.12) 式の右辺第2項以降と $\frac{E_X\cdot E_Y}{\hat{Y}^2}$ はかなり小さな値であるから捨象すれば，商 $\frac{X}{Y}$ は次のように書ける。

$$\frac{X}{Y}\simeq\frac{\hat{X}}{\hat{Y}}+\frac{E_X}{\hat{Y}}-\frac{\hat{X}}{\hat{Y}^2}E_Y \quad (5.13)$$

従って，商の絶対誤差は

$$E_{X/Y}\simeq\frac{1}{\hat{Y}}E_X-\frac{\hat{X}}{\hat{Y}^2}E_Y \quad (5.14)$$

商の相対誤差は次のようになる。

$$\frac{E_{X/Y}}{\hat{X}/\hat{Y}}\simeq\frac{E_X}{\hat{X}}-\frac{E_Y}{\hat{Y}} \quad (5.15)$$

すなわち，商の相対誤差は近似的に個別の相対誤差の差に等しい。

以上の 5.2～5.5 の各節で見出された誤差の合成の法則を用いて，四則演算を重ねて行なって行く時にどのような誤差累積が生ずるかを追跡するのが次の課題である。

6. 誤差累積の追跡

6.1. 四則演算結果の Rounding Error

(5.3), (5.5), (5.9), (5.15) の各式は，誤差を持った数値を加減乗除することによって合成された誤差の相対的大きさを示している。このような新しい誤差を持った数値を改めてコンピューターに記憶する場合に，再び新しい rounding error が発生する。この rounding error がどのように拡大するかをみるために，はじめは誤差を持たずに記憶された3つの数値X, Y, Zを想定して，これらの間で

$$U=(X+Y)\cdot Z \quad (6.1)$$

の計算例を考えてみる。但しコンピューターの有効桁数は t 桁とする。

最初の計算ステップ，X＋Yによって得られる答について生ずる切り捨て誤差の限界は，

$$\left|\frac{E_{X+Y}}{\hat{X}+\hat{Y}}\right|\leqq 10^{-t+1} \quad (6.2)$$

である。次に (X+Y) に Z を掛けて生ずる誤差は (5.9) 式に示した通り元の相対誤差の和であるから，全体の相対誤差は次のようになる。

$$\frac{E_U}{\hat{U}}=\frac{E_{X+Y}}{\hat{X}+\hat{Y}}+\frac{E_Z}{Z}+\gamma_c \quad (6.3)$$

但し，(6.3)式の右辺最終項の γ_c は，最終結果Uを記憶する際に発生する切り捨ての相対誤差である。

この例題では，Zは誤差を持たないと仮定しているから，これを除けば全体の誤差の範囲は

$$\left|\frac{E_u}{\hat{U}}\right|=\left|\frac{E_{X+Y}}{\hat{X}+\hat{Y}}+\gamma_c\right|\leqq\left|\frac{E_{X+Y}}{\hat{X}+\hat{Y}}\right|+|\gamma_c| \quad (6.4)$$

となる。E_{X+Y} と γ_c が同符号の場合は (6.4) 式の等号が成立ち異符号の場合は不等号が成立する。

ところで，4章で既述の通り，切り捨ての相対誤差の範囲は $|\gamma_c|\leqq 10^{-t+1}$ である。また，

$\left|\frac{E_{X+Y}}{\hat{X}+\hat{Y}}\right|\leqq 10^{-t+1}$ である。従って，(6.4) 式の最大限は下記のようになる。

$$\left|\frac{E_u}{\hat{U}}\right|\leqq 10^{-t+1}+10^{-t+1}=2\times 10^{-t+1} \quad (6.5)$$

(6.5) 式を書き換えれば

$$E_u\leqq|\hat{U}|\times 2\times 10^{-t+1} \quad (6.6)$$

であるから，$U=(X+Y)\cdot Z$ の絶対誤差は計算値 $\hat{U}$ の $2\times 10^{-t+1}$ 倍を超えることはない。

6.2 プロセス・グラフの方法

McCracken と Dorn は，多数の演算ステップを繰り返す場合について，(5.3), (5.5), (5.9), (5.15), (6.5) の各式で示した誤差合成がどのように波及し合って行くかを追跡するために，プロセス・グラフの方法を提案した。

プロセス・グラフを用いると，一連の演算ステップを矢印と演算記号で示すことによって，最終的な誤差の範囲を知ることができる。また，途中の演算ステップで発生する誤差が最終の誤差に与える影響を容易に知ることができる。プロセス・グラフは，次のような

ルールで書く。

(1) 演算プロセスは，下から上へ向って矢印で表示する。

(2) 水平方向にみて同じ高さの要素同士の計算を実行してから，一段上のレベルの計算に移る。

(3) 数値は，ⓧ，ⓨ，ⓩといったように記述する。

(4) 演算の記号は，⊕，⊖，⊙，⊘で加減乗除を表わす。

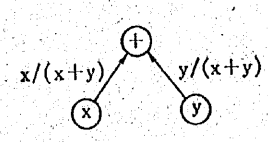
(5) 加算は，2つの数値(例えばⓧとⓨ)から⊕への矢印で表わす。2本の矢印には必ずラベルをつける。ⓧから⊕への矢印のラベルは，$x/(x+y)$，ⓨから⊕への矢印のラベルは$y/(x+y)$と記す約束とする。〔Fig. 1〕を参照されたい。

(6) 減算の場合は，〔Fig 2〕のように，ⓧから⊖への矢印には$x/(x-y)$のラベルを，ⓨから⊖への矢印には，$y/(x-y)$のラベルをつける。〔Fig. 2〕を参照。

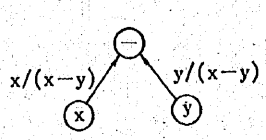
(7) 乗算の矢印には，+1というラベルをつける。〔Fig. 3〕を参照されたい。これはⓧのプラス1乗の意味である。

(8) 除算x/yの場合，ⓧから⊘への矢印(分子から⊘への矢印)には，xのプラス1乗の意味で，+1のラベルをつける。分母ⓨから⊘への矢印には，yのマイナス1乗の意味で，−1のラベルをつける。〔Fig. 4〕を参照。

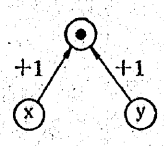
〔Fig. 1〕
$x+y$のプロセス・グラフ



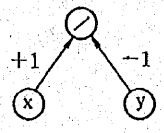
〔Fig. 2〕
$x-y$のプロセス・グラフ



〔Fig. 3〕
$x \cdot y$のプロセス・グラフ



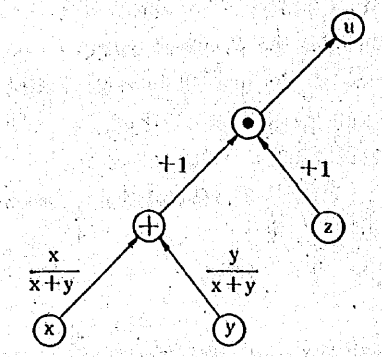
〔Fig. 4〕
x/yのプロセス・グラフ



(9) 個々の数値が持っている相対誤差をr_x，r_yのように表わす。1回の演算でこれらがどのように累積するかを知るには，ルール(5)，(6)，(7)，(8)で約束したラベルを掛けて加えればよい。

ルール(9)によって累積誤差が計算できる。〔Fig. 5〕をみながら，例題$u=(x+y)\cdot z$の誤差累積状況を見よう。

〔Fig. 5〕 $u=(x+y)\cdot z$のプロセス・グラフ



xとyは，始めに，それぞれr_x，r_yの相対誤差をもっているとする。演算の第1ステップ，$x+y$では，xからは$r_x\cdot\frac{x}{x+y}$，yからは$r_y\cdot\frac{y}{x+y}$の相対誤差が引き継がれる(5, 3式)を想起されたい。第1ステップの計算結果$(x+y)$を記憶する際に発生する切り捨ての相対誤差をr_1とすると，〔Fig. 5〕の下から2段目の演算＋が完了した時点までの誤差は，

$$\frac{E_{x+y}}{x+y}=\frac{x}{x+y}r_x+\frac{y}{x+y}r_y+r_1 \qquad (6.7)$$

次に第2ステップ$(x+y)\cdot z$へ進む。この乗算の一方の要素$(x+y)$は(6.7)式で示した誤差をもっている。一方の乗数zはr_zの相対誤差をもっているとしよう。また，最終結果uの記憶の際に発生する切り捨ての相対誤差をr_2とする。全体の相対誤差はルール(7)と(6.7)式から次のように評価できる。

$$\frac{E_u}{u}=\frac{E_{x+y}}{x+y}\times(+1)+r_z\times(+1)+r_2$$

$$=\frac{x}{x+y}r_x+\frac{y}{x+y}r_y+r_1+r_z+r_2 \qquad (6.8)$$

仮定したすべての誤差$(r_x, r_y, r_z, r_1, r_2)$が切り捨てによって発生したものであれば，各々の誤差は(4.5)式の限界を超えない筈であるから，全体の誤差の限界は次のようになる。

$$\left|\frac{E_u}{u}\right|\leqq\left(\left|\frac{x}{x+y}\right|+\left|\frac{y}{x+y}\right|+1+1+1\right)\cdot 10^{-t+1} \qquad (6.9)$$

特殊なケースとして，$x, y\geqq 0$ならば，

$\left|\frac{x}{x+y}\right|+\left|\frac{y}{x+y}\right|\leqq 1$であるから，(6.9)式はもっと具体的に

$$\left|\frac{E_u}{u}\right|\leqq 4\times 10^{-t+1}$$

という相対誤差の範囲を教えてくれることになる。

〔文 献〕

〔1〕 J. W. Longley, "An appraisal of least squares programs for the electronic computer from the point of view of the user," Journal of the American Statistical Association, Vol. 62, No. 319 819-841, September 1967.

〔2〕 Robert E. Hall, The Calculation of Ordinary Least Squares Estimates, Working paper No. 130, Mach 1968. University of California, Berkeley.

〔3〕 P. Davis, "Orthonormalizing Codes in Numerical Analysis," in J. Todd (ed.), Survey of Numerical Analysis (New York: McGraw-Hill, 1962).

鳥 居 泰 彦 (経済学部助教授)
新 井 益 洋 (産業研究所助手)