数値計算における誤差について
–数値微分を例に–

幸谷智紀
tkouya@na-net.ornl.gov

2005 年 9 月 21 日

更新履歴

Version 0.22, 2005 年 9 月 21 日 (水) 「経験的」の説明を付加。

Version 0.21, 2005 年 9 月 9 日 (金) 説明の微修正。

Version 0.2, 2005 年 8 月 29 日 (月) e の打ち切り誤差式の修正 (結論は変化なし)。

Version 0.1, 2005 年 7 月 9 日 (土) 「経験論的誤差推定法」についての記述を追加。

Version 0.0, 2005 年 6 月 13 日 (月) Web にて公開。

1 初めに

この短いレポートはある方から求められて書いたものである。最後に挙げてある例として数値微分を用いているのも，それに関する誤差解析についてのお尋ねであったためである。それも，なるべく簡便に丸め誤差と打ち切り誤差についての考え方と計測法を知りたい，ということであった。

実はその手の要請，とゆーか，淡い期待みたいなものは何度か聞いているのである。近年，特に日本の数値解析の中心テーマとして精度保証 [1] が注目されてくるにつれて，それとは反対に,「保証なんて厳密なもんじゃなくって，もっと簡便に誤差を計測したい」という要望も聞かれるようになってきたのである。

だったら，あの本 [3] とかこの本 [4] とか,「経験論的誤差推定法 (Empirical Error Estimation(E3)」とも言うべき，古典的だけど今でも通じる手法をまとめた本でも読めばいいじゃないか，ということになるのだが，The Internet 花盛りの昨今であるから,「丸め誤差と打ち切り誤差の計測法」だけを抜書きしたちょっとした文書が Google 検索で見つかればそれはそれで便利であろうし，そーゆー「しっかりし

た本」へのポインタとしても機能することが期待できる。ではいっちょ公開しようか，という，まあ「その程度」のものである。

二日ほどで慌ててまとめたものであるし，こちらもまだまだ浅学かつ非才の人間なので，間違い等あったらタイトルにあるメールアドレスまでお知らせ頂きたい。

2　丸め誤差とその計測法

2.1　丸め誤差とは

実数 $\mathbb{R}$ は，有限桁の小数もしくは無限桁の循環小数として表現できる有理数 $\mathbb{Q}$ と，循環しない無限小数としてしか表現できない無理数に分類される。

一方，コンピュータの記憶領域は有限であるため，有限桁の数しか扱うことができない。また，実用に耐える高速な計算を実現するには，なるべく桁は短いことが望ましい。しかし短い桁の小数しか扱えないと，後述する丸め誤差が大きくなり，数値計算の精度が悪くなる。このように，計算の高速性を維持しつつ，必要な数値計算の精度を保つため，実際の PC や WS では 2 進 24bits(符号部・指数部を加えて 32bits, IEEE754 単精度), 2 進 53bits(同 64bits, IEEE754 倍精度), 2 進 64bits(同 80bits, 拡張倍精度) の小数部を持つ浮動小数点数を用いることが多い。これらは 10 進数に換算して約 7 桁 ($\approx 24\log_{10}2$)，約 16 桁，約 20 桁の精度となる。

従って，実数をコンピュータ上で扱うには，無限桁の小数を有限桁の浮動小数点数に変換する操作が不可欠となる。この操作を丸め (round-off) と呼び，この丸めの結果生じる真の実数値とのずれを，丸め誤差 (round-off error) と呼ぶ。

例えば，円周率 π= 3.1415926535897932$\cdots$ を 10 進 7 桁の浮動小数点数に丸めてみる。浮動小数点形式にすると $\pm M.MMMMMM \times 10^E$ となる。

標準的な方法としては，小数部の一桁目が 1 の位になるよう指数部 E を決定した後，小数部の 8 桁目をチェックし，それが 4 以下ならそのまま切捨て，5 以上なら 7 桁目に 1 を加えて切捨てる，四捨五入法 (2 進数の場合は，0 捨 1 入) がある。この例では 3.141593 に丸められるので，相対丸め誤差は

$$\left|\frac{3.141593 - 3.141592653589\cdots}{3.141592653589\cdots}\right| \approx 1.1 \times 10^{-7}$$

となる。任意の実数に対して丸めによる相対誤差は $u = 1/2 \times 10^{-6}$ 以内に抑えられる。この u を丸め誤差の最小単位と呼ぶ。

コンピュータの内部における演算の結果が，指定された浮動小数点数の桁数に収まらない場合も丸めが必要となる。従って，浮動小数点演算ごとに丸め誤差は発生することになる。

例えば，自然対数の底 e= 2.7182818284590452$\cdots$ と π との積 $e\pi$ を 10 進 7 桁で計算する手順は次のようになる。

(1) e と π を 10 進 7 桁の浮動小数点数 $\tilde{e} = 2.718282$, $\tilde{\pi} = 3.141593$ に丸める。

(2) $\tilde{e} \times \tilde{\pi} = 8.539735703\cdots$ を計算する。

(3) (2) の結果を 10 進 7 桁 $\widetilde{\tilde{e}\tilde{\pi}} = 8.539736$ に丸める。

このように，(1) と (3) で 2 回の丸めが行われ，それぞれで丸め誤差が発生する。

大規模な科学技術計算では莫大な量の浮動小数点演算を実行するが，この多くで丸め誤差が発生する。最も悲観的な見方をすれば，全ての計算で u の丸め誤差が発生して蓄積し，最終結果は演算回数 N を乗じた Nu に達してしまうことになる。しかし，実際にここまで丸め誤差が増大することは極めて希であり，計算の途中で生じた丸め誤差が互いに打ち消しあって，せいぜい多めに見積もっても $\sqrt{N}u$ 程度になる，というのが大方の研究者の常識と考えてよい。この辺りは，かなりの部分,「経験」に依存して「常識」が固まってきた感がある。

2.2 丸め誤差の計測方法

実際の浮動小数点演算で生じた丸め誤差を計測するには，より長い桁の浮動小数点数を用いて同じ計算を行い，その結果との差を取るのが最も確実かつ簡易な方法である。

前述の $e\pi$ の計算例の場合，例えば 10 進 10 桁で同じ計算を行うことで，最終結果に含まれる丸め誤差 (の近似値) を計測することができる。実際

(1)′ e と π を 10 進 10 桁の浮動小数点数 $\tilde{e}' = 2.718281828$, $\tilde{\pi}' = 3.141592654$ に丸める。

(2)′ $\tilde{e}' \times \tilde{\pi}' = 8.539734222346491\cdots$ を計算する。

(3)′ (2)′ の結果を 10 進 7 桁 $\widetilde{\tilde{e}'\tilde{\pi}'} = 8.539734222$ に丸める。

となるから，これと $\widetilde{\tilde{e}\tilde{\pi}}$ との差を取り，真値の代わりに精度桁の多い方の結果で割って求めた

$$\left| \frac{\widetilde{\tilde{e}\tilde{\pi}} - \widetilde{\tilde{e}'\tilde{\pi}'}}{\widetilde{\tilde{e}'\tilde{\pi}'}} \right| \approx 2.1 \times 10^{-7}$$

が 10 進 7 桁計算で発生した相対丸め誤差であるということになる。

比較のために長い桁数の浮動小数点数が利用できない場合は，IEEE754 standard で定められている丸めモードを変更して区間演算 [5] を行ったり，より簡易な方法 [6] で推定することも可能であるが，詳細は長くなるので参考文献に譲ることにする。

3　打ち切り誤差とその計測法

3.1　打ち切り誤差とは

丸め誤差が実数を有限桁の浮動小数点数で近似した結果生じた誤差であるのに対し，打ち切り誤差は無限級数や極限値のような無限回の演算を必要とする解析表現を，有限回の演算で打ち切る (truncate) ことによって生じる誤差である。丸め誤差は数値によって変動し予測が難しいのに対し，打ち切り誤差は解析表現が明らかであれば，それに基づいて予測することが可能である。故に，打ち切り誤差は理論誤差とも呼ばれる。

例えば，e は指数関数 $\exp(x)(= e^x)$ の Maclaurin 展開式によって

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!} + \cdots$$

という無限級数の形で表現される。しかし，いかに高速なコンピュータといえども無限級数を計算することはできないため，どこかの項 $1/m!$ で計算を打ち切る必要がある。この項までの有限和を $\hat{e}_m$ と書くことにする。すなわち，

$$\hat{e}_m = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{m!}$$

である。この時，打ち切り誤差は

$$e - \hat{e}_m = \frac{1}{(m+1)!} + \frac{1}{(m+2)!} + \cdots$$

となる。

3.2　打ち切り誤差の計測法と丸め誤差との関連

この形では評価が難しいので有限和の Maclaurin 展開公式

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{m!} + \frac{\exp(\theta)}{(m+1)!}$$

を用いることにする。ここで θ は $0 < \theta < 1$ となる定数である。

これを用いると打ち切り誤差は

$$e - \hat{e}_m = \frac{\exp(\theta)}{(m+1)!}$$

となる。右辺の絶対値を取れば

$$\left| \frac{\exp(\theta)}{(m+1)!} \right| \leq \frac{e}{(m+1)!}$$

となるので，相対打ち切り誤差を取ると

$$\left|\frac{e-\hat{e}_m}{e}\right| \leq \frac{1}{(m+1)!} \tag{1}$$

となり，m が決まれば打ち切り誤差の上限を評価することが可能となる。

一般に，打ち切り誤差は計算回数さえ増やせば減らすことができるが，使用する浮動小数点数の丸め誤差の最小単位より過度に小さくしても，コンピュータ資源の無駄遣いにしかならない。

例えば，10 進 7 桁の浮動小数点数を用いて e を計算するのであれば，先の評価式 (1) を用いて

$$\left|\frac{e-\hat{e}_m}{e}\right| \leq \frac{1}{(m+1)!} \approx \frac{1}{2}\cdot 10^{-6}$$

程度になる m まで計算するのが最適と言える。この場合，

$$\frac{1}{(8+1)!} \approx 2.8\times 10^{-6},\ \frac{1}{(9+1)!} \approx 2.8\times 10^{-7}$$

であるから，$m=9$，すなわち

$$\hat{e}_9 = 1+\frac{1}{1!}+\frac{1}{2!}+\cdots+\frac{1}{9!}$$

程度まで計算しておけば十分である。実際に計算してみると

$$\hat{e}_9 = \underline{2.71828152557}\cdots$$

であり，下線部の 7 桁分が真値と一致していることが分かる。

4　数値微分の経験論的誤差推定法

一変数関数 $f(x)$ の，$x=a$ における微係数 $f'(a)$ 対する，Stirling の中心差分公式に基づく数値微分の近似式 (下線部) および打ち切り誤差は次のようになる [2]。

$$\begin{aligned}
\text{[3 点公式] } f'(a) &= \underline{\frac{1}{h}\left\{\frac{1}{2}f(a+h)-\frac{1}{2}f(a-h)\right\}} \\
&+ \frac{1}{6}f^{(3)}(a)h^2+\cdots \qquad (2)\\
\text{[5 点公式] } f'(a) &= \underline{\frac{1}{h}\left\{-\frac{1}{12}f(a+2h)+\frac{2}{3}f(a+h)-\frac{2}{3}f(a-h)+\frac{1}{12}f(a-2h)\right\}} \\
&+ \frac{1}{30}f^{(5)}(a)h^4+\cdots \qquad (3)\\
\text{[7 点公式] } f'(a) &= \underline{\frac{1}{h}\left\{\frac{1}{60}f(a+3h)-\frac{3}{20}f(a+2h)+\frac{3}{4}f(a+h)-\frac{3}{4}f(a-h)\right.} \\
&+ \underline{\left.\frac{3}{20}f(a-2h)-\frac{1}{60}f(a-3h)\right\}} \\
&+ \frac{1}{140}f^{(7)}(a)h^6+\cdots \qquad (4)
\end{aligned}$$

ここで，h は刻み幅と呼ばれる定数で，通常はなるべく小さい値になるように取る。どの公式でも打ち切り誤差はこの刻み幅の多項式として表現されていることがわかる。この場合，誤差項の最小次数の order で打ち切り誤差を見積もることができるため，h を小さくすれば，3 点公式は h^2 に，5 点公式は h^4 に，7 点公式は h^6 に比例して打ち切り誤差が縮小していくと予測できる。

実際に，これらの数値微分公式を用いた数値実験を行い，丸め誤差と打切誤差の計測を行ってみる。関数は

$$f(x) = \cos(\sin x)$$

を使用する。この導関数は

$$f'(x) = -\sin(\sin x) \cdot \cos x$$

であることが分かっているので，この関数値を微係数の真値として使用する。評価するのは $x = \pi/4$ における微係数 $f'(\pi/4) = -4.593626849327 \times 10^{-1}$ である。

まず，刻み幅 h を $2^0, 2^{-1}, \ldots, 2^{-10}$ と小さくしていった時に発生する相対打ち切り誤差 (TE) と，相対丸め誤差 (RE) をそれぞれ計測してみる。その結果が図 1 であり，縦軸に相対誤差 (対数スケール)，横軸に $h = 2^{-x}$ を取っている。打ち切り誤差は 10 進 100 桁計算の結果を，丸め誤差は IEEE754 倍精度の計算結果と拡張倍精度の計算結果との相対差を取って，それぞれプロットしてある。

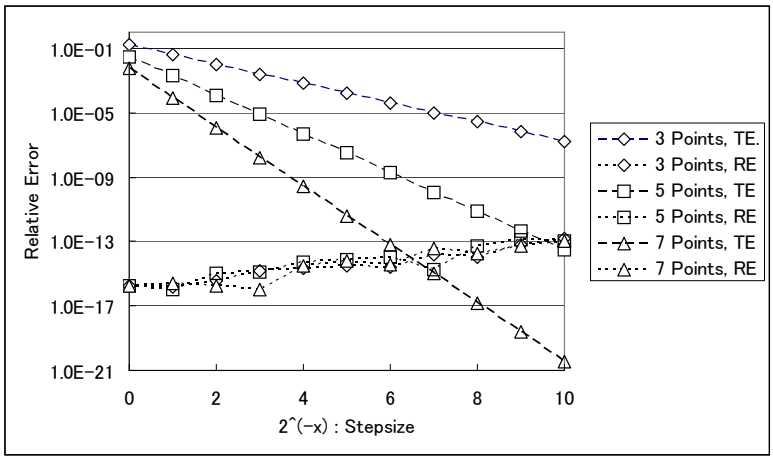

図 1: $x = \pi/4$ における打ち切り誤差 (TE) と丸め誤差 (RE)

刻み幅を小さく (グラフの右方向) していくと，予想通り 3 点, 5 点, 7 点公式の打ち切り誤差はそれぞれ h^2, h^4, h^6 に比例して小さくなっていることがわかる。これにより，打ち切り誤差の計測を行うには，一つの刻み幅を用いた計算だけでは不十分で，少なくとも三つ以上の異なる刻み幅 h を用いて計算した結果を比較する必要があることが分かる。

一方，丸め誤差は，同じ IEEE754 倍精度の浮動小数点数を用いているため，どの公式も大差なく推移している。数値微分においては，刻み幅 h を小さくしていくと必然的に桁落ちを伴うため，相対丸め誤差が若干高くなる傾向が見られる。

IEEE754 倍精度で計算した数値微分の数値結果に含まれる誤差は，図 1 に示した打ち切り誤差と丸め誤差を合わせたものとなる。従って，例えば 3 点公式を用いた場合は，$h = 2^0$ から 2^{-10} の刻み幅ではいずれも打ち切り誤差の方が丸め誤差を上回っているため，数値結果に含まれる誤差は打ち切り誤差のみが顕在化していると予想できる。逆に，7 点公式の場合は，ちょうど $h = 2^{-6}$ の地点で打ち切り誤差と丸め誤差の大きさが拮抗しているため，これより大きい刻み幅では打ち切り誤差が，小さい刻み幅では丸め誤差が優越してくると予想できる。これらの中間の打ち切り誤差を持つ 5 点公式は，ちょうど $h = 2^{-10}$ が打ち切り・丸め誤差の拮抗点となっている。

IEEE754 倍精度を用いて計算した結果の相対誤差を取ったものが図 2 である。以上の予想が正しいことが見て取れる。

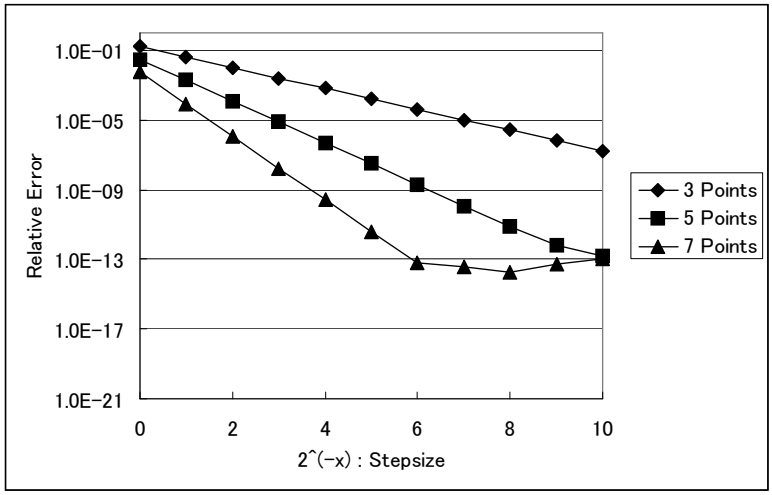

図 2: $x = \pi/4$ における相対誤差

現実の計算では，真の微係数が不明である故に数値微分を行うのが普通である。よって，数値結果のみから誤差の推定を行う必要があるが，以上の結果から，複数の刻み幅を用いて計算した値を比較することで，おおよその予測ができることになる。

今回使用した $f'(\pi/4)$ を 7 点公式を用いて計算した値は表 1 のようになる。それぞれの刻み幅を用いて計算した数値のうち，一段階大きい刻み幅の数値と上の位の桁から比較して一致している部分に下線を引いてある。図 2 と比較して見て頂きたい。

刻み幅を $h = 2^{-6}$ まで小さくしていくと，一致していく桁数が一定して増大していくことが見て取れる。これは打ち切り誤差が減少していく部分に相当する。一

表 1: 7 点公式による近似値

刻み幅	7 点公式で得た $f'(\pi/4)$ の値
2^{0}	−4.56886082650315217e − 01
2^{-1}	−4.59327000065456403e − 01
2^{-2}	−4.59362179303235640e − 01
2^{-3}	−4.59362677301507094e − 01
2^{-4}	−4.59362684814669853e − 01
2^{-5}	−4.59362684930946064e − 01
2^{-6}	−4.59362684932753673e − 01
2^{-7}	−4.59362684932801912e − 01
2^{-8}	−4.59362684932793197e − 01
2^{-9}	−4.59362684932760557e − 01
2^{-10}	−4.59362684932832721e − 01

方，刻み幅を 2^{-6} より小さくしても一致している桁数は増えず，−4.593626849327 と −4.593626849328 の間を揺れ動いている。これは丸め誤差が打ち切り誤差を上回ったことを示している。よって，これ以上刻み幅を小さくしても相対誤差の減少は望めそうも無く，真値は $-4.59362684933 \times 10^{-1}$，この精度は約 12 桁 ••• と結論付けられる。

このようにして，複数の刻み幅を用いた計算結果を比較することで，真の値が不明であっても，ある程度の相対誤差の「あたり」をつけることは可能である [3]。

ここでは特定の $x = a$ 地点における数値結果の相対誤差を推定したが，ある区間全体にこの結果が敷衍できるかどうかは関数自身の性質に依存するので一概には言えない。今回使用した関数は $[-2\pi, 2\pi]$ で周期関数となり，この区間内で無限回連続微分可能であり，その導関数の変動幅もごく小さいことから，桁落ちが起きる地点を除けば，どの地点においても同じような精度の数値結果を得ることができると推察される。解析的な性質がこのようなごく「おとなしい」ものであるような関数及び区間においても同様のことが言える。

5　経験論的誤差推定法の信憑性は？

以上，数値微分の計算における打ち切り誤差及び丸め誤差の推定法について述べたが，厳密に言うと，この手法は合理的な考え方に基づくものだが，あくまで経験的なものでしかなく,「絶対確実」なものではないことをお断りしておく。しかしそれでも,「かなりの割合」でこの経験則に基づく誤差推定法は信憑性がある，ということは歴史が証明しているし，研究者の間では「使える」手法として常識

化しているのである。
　以上を踏まえた上で，今，自分の目の前で実行された近似計算の誤差推定をどのように実行するかを判断されたい。ま，自己責任って奴ですかね。

参考文献

[1] 大石進一, 精度保障付き数値計算, コロナ社, 2000.

[2] 永坂秀子, 計算機と数値解析, 朝倉書店, 1980.

[3] 伊理正夫・藤野和建, 数値計算の常識, 共立出版, 1985.

[4] 二宮市三編, 数値計算のつぼ, 共立出版, 2004.

[5] A. Neumaier, Interval Methods for Systems of Equations, Cambridge University Press, 1990.

[6] 幸谷智紀・永坂秀子, IEEE754 規格を用いた簡易な丸め誤差の計測法について, 日本応用数理学会論文誌, Vol.7, No.1.