

高速自動微分法と区間解析とを用いた丸め誤差推定

久保田 光一, 伊理 正夫
東京大学工学部計数工学科

本稿では, 関数の計算値に含まれる丸め誤差の絶対値の厳密な上界を求める実用的な算法を提案する. また, 普通の条件下では, 演算の精度を高めるに従い, その上界が上限に近づくことも述べる. 我々の手法は, 浮動小数点数を両端点に持つ実数区間の区間演算 (機械区間演算と呼ぶ) と “高速自動微分法” とを組合せたものである. 高速自動微分法は, 多変数の関数の値を計算する手続き (プログラム) が与えられたとき, 関数値の計算に必要な手間の高々定数倍の手間で, 関数のすべての変数に関する偏導関数値および関数の計算値に含まれる丸め誤差の絶対値の良い上界を計算する実用的な方法である.

本稿の手法によれば, 丸め誤差の絶対値の厳密な上界と数値計算結果とから, 丸め誤差のない真の関数値を含む区間を定めることが可能となり, 数値計算結果の品質の保証につながる. 通常の区間解析によってもそのような保証区間を計算できるが, 大規模・複雑な関数に対しては, その区間幅が非実用的に過大評価されがちである. 線形方程式系の解に含まれる丸め誤差を例にとって, 区間解析と比較して, 本手法の有効性を数値実験的に示す.

Estimates of Rounding Errors with Fast Automatic Differentiation and Interval Analysis

Koichi KUBOTA and Masao IRI
Department of Mathematical Engineering and Information Physics,
Faculty of Engineering, University of Tokyo,
Bunkyo-ku, Tokyo 113, Japan.

We propose an algorithm for calculating rigorous upper bounds of the absolute values of the rounding errors incurred in the computed values of functions. We also prove that, under some suitable conditions, the upper bounds become asymptotically sharp as the machine epsilon approaches zero. The proposed method is a combination of the technique of Fast Automatic Differentiation and interval operations for real intervals with both upper and lower edges representable as floating-point numbers.

The method is useful to guarantee the quality of numerical results in the sense that we can determine an interval containing the exact value of the function based on the value computed in finite precision and on the estimate obtained by the method. We also illustrate the efficiency and the practicalness of the proposed method by numerical experiments in comparison with the conventional interval analysis.

1. まえがき

高速自動微分法^{4),5),6),8)}を用いることによって、我々は、実用的な手間で、関数の計算値に含まれる丸め誤差を推定できるようになった。既に、丸め誤差の「線形近似」に基づき「絶対評価」と「確率評価」と呼ばれる2種類の推定方法が提案され、これらの方法による丸め誤差の推定値が実際に発生する丸め誤差の大きさの上界を良く近似していることが実験により確かめられている⁶⁾。しかし、「確率評価」はもちろん「絶対評価」も丸め誤差の絶対値の“厳密な”上界ではないために、これらの推定値を用いて真の値の存在範囲を完全に保証することはできない。

従来、数値計算の結果の品質を保証するためには、関数を計算する際に実行される各演算をいわゆる機械区間演算に置き換えて計算を行うこと以外には、実際のな方法がなかった。その結果として得られる区間は、真の値を“厳密に”含み、真の値の存在範囲(区間)を保証する。しかし、以下に実例で示すように、関数の規模が大きくなると、この区間の幅は桁違いに大きくなる場合がある。そこで、我々は、このような「計算結果を保証する」という立場から、高速自動微分法と区間解析とを組み合わせると、丸め誤差の絶対値の厳密な上界で、かつ、後で述べる意味で最適な上界であるような丸め誤差推定値を求めることができることを示し、そのための具体的な算法を提案する。さらに、その算法による結果を、単なる区間演算の結果および線形近似による絶対評価と実験的に比較して、実用的観点からの評価を加える。

以下では、関数に与える変数の値には誤差が無いと仮定し、計算による誤差だけに注目する。(もちろん、変数の値に誤差が含まれている場合への拡張も容易である。)さらに、我々は次のような状況を考える: 「各演算は与えられた一定語長内で最大精度で遂行され、我々には発生誤差の絶対値がある大きさ以下であるということだけが判り、その符号や大きさは知ることができない。」これは、丸め誤差を厳密に論じる際の最も妥当な状況設定であろう。

我々が以下で行うことは、微分の平均値の定理を利用した Yu. V. Matiyasevich による丸めの無い区間解析の方法⁹⁾を丸め誤差推定も扱えるように拡張することである。区間解析に微分の平均値の定理を利用することは、E. R. Hansen 等によって述べられていた^{3),10)}。Matiyasevich は Hansen の方法の計算の手間を改良した。(Hansen の方法に必要な手間は、関数の変数の数に比例して増加する。Matiyasevich の方法に必要な手間は、高速微分法を利用しているので、変数の数に関係なく、関数を計算する手間だけに比例する。)

第2節で、取り扱い関数の範囲と丸め誤差を伴う計算について説明し、第3節では機械区間演算、算法、最適性等について述べる。第4節では、数値実験結果について述べる。

2. 丸め誤差を伴う計算

2.1. 分解可能関数と計算過程

関数を計算する過程で使用することのできる演算を $+$ 、 $-$ 、 \times 、 $/$ 、 \exp 、 \log などの(その定義域においては)一階連続微分可能な単項または2項演算に限定し、これらを「基本演算」と呼ぶ。本論文で取り扱う

$$\begin{aligned} v_1 &\leftarrow \psi_1(u_{11}, u_{12}) \\ &\vdots \\ v_j &\leftarrow \psi_j(u_{j1}, u_{j2}) \\ &\vdots \\ (f =) v_r &\leftarrow \psi_r(u_{r1}, u_{r2}) \end{aligned}$$

v_1, \dots, v_r : 中間変数;
 ψ_1, \dots, ψ_r : 基本演算;
 u_{j1}, u_{j2} : ψ_j の引数.
(入力変数 x_1, \dots, x_n , 定数, 中間変数 v_1, \dots, v_{j-1} のいずれかがこれらに対する実引数となる ($j = 1, \dots, r$.)

図1. 計算過程

関数は、基本演算の組合せによって関数値を計算できる分解可能関数・区分的分解可能関数⁷⁾と呼ばれるもので、関数の変数(入力)の値から関数値を計算する手続きとして記述されるようなものであるとする(区分的分解可能関数の手続きは、条件分岐や反復を含んでもよい)。入力の値を定めて関数の手続きにしたがって計算すれば、我々は関数値を得ることができる。その計算で実行された基本演算の履歴を「計算過程」と呼ぶ。計算過程は、基本演算を1回遂行してその結果を中間変数に保持するという操作(計算ステップ)の列である。 n 変数関数 $f(x_1, \dots, x_n)$ の計算過程は図1のように表される。(なお、以下では、基本演算が2項演算の場合を示すが、単項演算に関しては第2引数に関係する部分を削除することと約束する。)中間変数の数は計算ステップの個数(以下では r と記す)に等しい。

次節以降では、与えられた入力に対する区分的分解可能関数の計算過程が得られていると仮定して、その計算過程を考察の対象とする。もちろん、区分的分解可能関数の手続きにしたがって計算する場合には、入力の値を変えると、条件分岐の分岐先や反復の回数などが変化することがあり、計算過程の構造—計算ステップの引数と実引数の関係や計算ステップの総数など—も変化することがある。しかし、入力の値が決まれば、誤差の無い計算(無限精度計算)の履歴としての区分的分解可能関数の計算過程は一意に定まる。(分解可能関数の計算過程の構造は入力数値に関わらずに定まるので、分解可能関数の議論は区分的分解可能関数の議論に含まれる。)

2.2. 浮動小数点方式と誤差のある計算

通常、計算機では浮動小数点演算によって実数演算を近似するが、このとき使用される浮動小数点数の表現および丸めの方式を合わせて浮動小数点方式と呼ぶことにする。ほとんどの計算機では、マシンエプシロン ϵ_M によって、浮動小数点方式の丸めの際の相対誤差の上限が表現される。

以下では、有限精度(すなわち誤差のある)計算による計算過程の構造は無有限精度計算による計算過程の構造と同じであるということ—有限精度演算による関数値計算の実行可能性—を仮定する。このような仮定をする理由は、有限精度計算を行うと、たと

$$\begin{aligned} \bar{v}_1 &\leftarrow \bar{\psi}_1(\bar{u}_{11}, \bar{u}_{12}) = \psi_1(\bar{u}_{11}, \bar{u}_{12}) + \delta_1 \\ &\vdots \\ \bar{v}_j &\leftarrow \bar{\psi}_j(\bar{u}_{j1}, \bar{u}_{j2}) = \psi_j(\bar{u}_{j1}, \bar{u}_{j2}) + \delta_j \\ &\vdots \\ (\bar{f} =) \bar{v}_r &\leftarrow \bar{\psi}_r(\bar{u}_{r1}, \bar{u}_{r2}) = \psi_r(\bar{u}_{r1}, \bar{u}_{r2}) + \delta_r \end{aligned}$$

$\bar{v}_1, \dots, \bar{v}_r$: 中間変数;
 $\bar{\psi}_1, \dots, \bar{\psi}_r$: 基本演算;
 $\bar{u}_{j1}, \bar{u}_{j2}$: ψ_j の引数;
 (入力変数 x_1, \dots, x_n , 定数, 中間変数 $\bar{v}_1, \dots, \bar{v}_{j-1}$ のいずれかがこれらに対する実引数となる ($j = 1, \dots, r$).)
 $\delta_1, \dots, \delta_r$: 発生誤差.

図 2. 丸めつき計算過程

え入力の値が同じであっても、計算過程の構造が無限精度計算によるものと異なることがありうるからである。すなわち、中間変数や関数の有限精度計算値と無限精度計算値の相違が、関数を表現する手続きの中で、条件分岐における分岐先の変化や符号が確定しない中間結果を除数とする除算などを引き起こし、計算の履歴である計算過程の構造の変化を生み出しうるからである。逆に、計算過程の構造が変化するということは、その入力数値と計算精度に関する限り、関数値が確定しない—計算不可能である—ということの意味する。それ故、我々の仮定は、その入力数値と計算精度に関する関数値計算の実行可能性の仮定であると言える。実際には誤差の無い計算の遂行は不可能であるので、誤差の無い計算過程の構造を直接知ることはできないが、第 3 節で述べるように、我々は、区間解析の手法により、上の仮定が成立していることを確かめることができる。(ただし、その手法は、上の仮定が成立するための十分条件を与えるだけであり、その入力数値と計算精度に関する関数値計算が実行不可能であることを直接示すのではない。)

浮動小数点演算を用いて、 f の手続きに沿って計算する場合には、真の値を浮動小数点数で近似するために誤差が生じる。上の仮定により、誤差が存在しても計算過程の構造は変化しないので、誤差を伴う演算を明示するために図 1 の第 j 番目の計算ステップ「 $v_j \leftarrow \psi_j(u_{j1}, u_{j2})$ 」の代わりに「 $\bar{v}_j \leftarrow \bar{\psi}_j(\bar{u}_{j1}, \bar{u}_{j2})$ 」と表すことにする。 $\bar{\psi}_j$ は有限精度の演算である(上線は有限精度の変数・演算などであることを示す)。引数と実引数の対応関係は図 1 のままであり、変わらない。第 j 番目の計算ステップに付随する“発生誤差” δ_j を、無限精度演算 ψ_j と有限精度演算 $\bar{\psi}_j$ とを比較して、

$$\bar{\psi}_j(\bar{u}_{j1}, \bar{u}_{j2}) = \psi_j(\bar{u}_{j1}, \bar{u}_{j2}) + \delta_j \quad (2.2.1)$$

によって定義する(図 2)。(2.2.1)の右辺 ψ_j の引数は左辺と同じく有限精度の計算結果であるので、発生誤差 δ_j は $\bar{\psi}_j$ の計算を行う段階で発生する局所的な誤差である。基本演算が連続微分可能であるので、第 j 番目の計算ステップの計算値 \bar{v}_j と真の値 v_j との差は

$$\begin{aligned} \bar{v}_j - v_j &= \bar{\psi}_j(\bar{u}_{j1}, \bar{u}_{j2}) - \psi_j(u_{j1}, u_{j2}) \\ &= \psi_j(\bar{u}_{j1}, \bar{u}_{j2}) - \psi_j(u_{j1}, u_{j2}) + \delta_j \\ &= \frac{\partial \psi_j}{\partial u_{j1}}(u_{j1} + \theta_j \cdot (\bar{u}_{j1} - u_{j1}), u_{j2} + \theta_j \cdot (\bar{u}_{j2} - u_{j2})) \\ &\quad \cdot (\bar{u}_{j1} - u_{j1}) \\ &\quad + \frac{\partial \psi_j}{\partial u_{j2}}(u_{j1} + \theta_j \cdot (\bar{u}_{j1} - u_{j1}), u_{j2} + \theta_j \cdot (\bar{u}_{j2} - u_{j2})) \\ &\quad \cdot (\bar{u}_{j2} - u_{j2}) + \delta_j \end{aligned} \quad (2.2.2)$$

である ($0 < \theta_j < 1$)。 $\partial \psi_j / \partial u_{j1}$, $\partial \psi_j / \partial u_{j2}$ は基本演算 ψ_j の偏導関数である。我々はこれらを要素的偏導関数^{4),5),6)}と呼び、その $(u_{j1} + \theta_j \cdot (\bar{u}_{j1} - u_{j1}), u_{j2} + \theta_j \cdot (\bar{u}_{j2} - u_{j2}))$ における値を

$$d_i^j \equiv \frac{\partial \psi_j}{\partial u_{ji}}(u_{j1} + \theta_j \cdot (\bar{u}_{j1} - u_{j1}), u_{j2} + \theta_j \cdot (\bar{u}_{j2} - u_{j2})) \quad (i = 1, 2). \quad (2.2.3)$$

と記す。これを用いると、(2.2.2)は

$$\bar{v}_j - v_j = d_1^j \cdot (\bar{u}_{j1} - u_{j1}) + d_2^j \cdot (\bar{u}_{j2} - u_{j2}) + \delta_j \quad (2.2.4)$$

と表せる。関数の計算値に含まれる丸め誤差は、最終結果 $\bar{v}_r (= \bar{f})$ と $v_r (= f)$ との差である：

$$\bar{f} - f = \bar{v}_r - v_r = d_1^r \cdot (\bar{u}_{r1} - u_{r1}) + d_2^r \cdot (\bar{u}_{r2} - u_{r2}) + \delta_r. \quad (2.2.5)$$

次に、我々は w_j ($j = 1, \dots, r$) を定義する。そのために、計算グラフ^{5),8)}, $G = (V, E, \partial^+, \partial^-, \omega, n, d)$, の記号を使用する。頂点は、中間変数(または、入力変数、定数)に対応し、枝 e は基本演算の仮引数に対応し、その基本演算は終点 ($\partial^- e$) に対応する中間変数を左辺に持つ計算ステップにおけるものである。始点 ($\partial^+ e$) はその実引数に対応する。計算グラフ上の v_j を始点とし v_m を終点とする有向道(長さは l とする) e_1, \dots, e_l ($\partial^+ e_1 = v_j$, $\partial^- e_k = \partial^+ e_{k+1}$ ($k = 1, \dots, l-1$), $\partial^- e_l = v_m$) に関して、 $\partial^- e_k$ に対応する中間変数の添字を s_k で表わし ($v_{s_k} = \partial^- e_k$), v_{s_k} を左辺に持つ計算ステップで枝 e_k が対応する仮引数の番号を i_k ($= 1$ または 2) と表すことにすると、 $\{(s_k, i_k)\}_{k=1}^l$ という列が定められる。第 j 番目の中間変数 v_j に対応する頂点を始点とし、関数値 v_r に対応する頂点を終点とする有向道を考え、各々に対応する $\{(s_k, i_k)\}_{k=1}^l$ という列の集合を Q_j と記す。我々は、 w_j を次のように定義する：

$$\begin{aligned} w_j &\equiv \sum_{\{(s_k, i_k)\}_{k=1}^l \in Q_j} d_{i_1}^{s_1} \cdot d_{i_2}^{s_2} \cdot \dots \cdot d_{i_l}^{s_l} \\ &= \left(\sum_{\{(s_k, i_k)\}_{k=1}^l \in Q_j} \frac{\partial \psi_r}{\partial u_{r i_1}} \cdot \frac{\partial \psi_{s_1}}{\partial u_{s_1 i_1}} \cdot \dots \cdot \frac{\partial \psi_{s_1}}{\partial u_{s_1 i_1}} \right) \end{aligned} \quad (2.2.6)$$

($j = 1, \dots, r-1$). ただし、 $w_r = 1$ とする。これらを用いて、(2.2.5)の $\bar{u}_{r1} - u_{r1}$, $\bar{u}_{r2} - u_{r2}$ を実引数に置き換えて(2.2.4)により順次展開すると

$$\bar{f} - f = \sum_{j=1}^r w_j \cdot \delta_j \quad (2.2.7)$$

が得られる。次節では、以上の議論を踏まえて、我々は、 $|\bar{f} - f|$ の上界を厳密に評価する算法について述べる。

一方、 $\theta_1 = 0, \dots, \theta_r = 0$ のときには、合成関数の微分の規則そのものによって、

$$w_j = \frac{\partial f}{\partial v_j} \quad (2.2.8)$$

である。「絶対評価」、 $|\bar{f} - f|$ の基礎になる線形近似 L_f は、各要素的偏導関数の値が θ_j に関わらず一定とみなすものである：

$$\bar{v}_r - v_r \simeq L_f \equiv \sum_{j=1}^r \frac{\partial f}{\partial v_j} \cdot \delta_j. \quad (2.2.9)$$

我々は、高速自動微分法により、実用的な手間で $\partial f / \partial v_j$ ($j = 1, \dots, r$) を計算できる (ただし、この計算も丸め誤差を含む)。ほとんどの計算機では $|\delta_j| \leq |v_j| \cdot \varepsilon_M$ (ε_M : マシンエプシロン) が成立することから、

$$A_f \equiv \sum_{j=1}^r \left| \frac{\partial f}{\partial v_j} \right| \cdot |v_j| \cdot \varepsilon_M \quad (2.2.10)$$

と表される絶対評価 A_f は、丸め誤差の絶対値の上界の実用的には十分によい近似であるが^(4),5),6),8)、線形近似 L_f が既に丸め誤差の近似であるので、厳密な上界とはいえない。

3. 丸め誤差の厳密な上界推定算法

$\bar{f} - f \in S$ なる区間[†] $S = [s^l, s^h]$ を計算する算法を以下に記す。我々は、 $|\bar{f} - f| \leq |S|$ によって丸め誤差の絶対値の厳密な上界を得ることができる。(ここで、区間 $X = [x^l, x^h]$ に対して $|X| \equiv \max\{|x^l|, |x^h|\}$ である。)

3.1. 区間演算・機械区間演算

実数の (閉) 区間 $A = [a^l, a^h]$ を単に「区間」と呼び、両端点 a^l, a^h がともに機械により表現可能な浮動小数点数であるような実数区間を「機械区間」と呼ぶことにする。区間演算とは、区間を引数とし、対応する基本演算による引数の区間の像 (これもまた区間となる) を結果とする演算である。機械区間演算は、機械区間を引数とし、かつ、その機械区間を引数とする (実数) 区間演算の結果を含むような幅が最狭の機械区間[‡] を結果とする演算である。もちろん、機械区間および機械区間演算は浮動小数点方式に依存する。¹⁾

f を計算する手続き (計算過程ではない!) に現れる全ての基本演算を対応する機械区間演算に置き換え、すべての変数を機械区間変数に置き換えることによって、機械区間を計算する手続きを作ることができる。入力として、幅が 0 の機械区間 $[\bar{x}_1, \bar{x}_1], \dots, [\bar{x}_n, \bar{x}_n]$ を与え、この手続きを遂行して得られる区間を \bar{F} とする (F は f と \bar{f} を両方とも含む機械区間であり、 \bar{F} の区間幅も丸め誤差の厳密な上界である

[†] 実数区間の表現方法には、上下両端点によるもの、中心および半径によるものなどがあるが、以下では上下両端点による表現を用いる。

[‡] あるいはその次に幅が狭い機械区間

が、 f の計算ステップの数が増えると \bar{F} の幅は一般には甚だしい過大評価を与える。) このときの計算の履歴は機械区間演算による計算過程である。

\bar{F} が計算できる — 零を含む機械区間を除くとする除算や共通部分を持つ機械区間の大小比較などが発生せずに機械区間演算の実行ができる — ことによって、その浮動小数点形式と入力数値とに関して §2.2 で述べた仮定 — 関数値計算の実行可能性 — の成立が保証される。何故ならば、浮動小数点方式を決めたとき、入力数値から \bar{F} を計算できる — 計算過程が得られる — という事は、誤差によって中間変数や関数の値が変化してもそれらの値は機械区間演算による中間変数や関数の機械区間に含まれ、計算過程の構造が変化しないということの意味するからである。逆に、 \bar{F} の計算ができない場合には、そのときの浮動小数点方式と入力数値に関して、関数は丸め誤差に弱いことがわかる。(マシンエプシロン ε_M の浮動小数点方式の機械区間演算で \bar{F} が計算できるとすれば、その浮動小数点方式よりも仮数部長・指数部長が長く、 ε_M よりも小さいマシンエプシロンで表される浮動小数点方式を用いた機械区間演算の結果は、必ず、 \bar{F} に含まれる。)

3.2. 偏微分係数の計算と丸め誤差の推定

まず機械区間演算を行って \bar{F} を得て関数値計算の実行可能性を確認してから、以下のようにして \bar{F} より幅の狭い区間を計算する。

\bar{F} の計算過程における f の中間変数 v_j 等に対応する機械区間を \bar{V}_j 等と表すと、

$$v_j, \bar{v}_j \in \bar{V}_j \quad (j = 1, \dots, r) \quad (3.2.1)$$

であり、これらが引数として現れるときには、

$$u_{ji} + \theta_j \cdot (\bar{u}_{ji} - u_{ji}) \in \bar{U}_{ji} \quad (j = 1, \dots, r; i = 1, 2) \quad (3.2.2)$$

という形をとる (ここで、 U_{ji} は引数 u_{ji} に対応する機械区間を示す)。

つぎに、我々は要素的偏導関数 $\partial \psi_j / \partial u_{ji}$ を計算するための演算を機械区間演算に置き換える。その機械区間演算の引数として u_{j1}, u_{j2} の実引数に対応する中間変数などの機械区間 (\bar{V}_k, \bar{V}_l など) を与えて計算される機械区間を \bar{D}_i^j とすれば、 $0 \leq \theta_j \leq 1$ であることから、

表 1. 要素的偏導関数に対応する機械区間

\bar{D}_i^j は $\partial \psi_j / \partial u_{ji}$ に対応する。
ただし、 $v_j = \psi_j(u_{j1}, u_{j2})$ $\left| \begin{matrix} u_{j1} = v_k, u_{j2} = v_l \end{matrix} \right.$

ψ_j	\bar{D}_1^j	\bar{D}_2^j
\pm	$[1, 1]$	$[\pm 1, \pm 1]$
$*$	\bar{V}_l	\bar{V}_k
$/$	$[1, 1] \odot \bar{V}_l$	$[-1, -1] \odot \bar{V}_j \odot \bar{V}_l$
exp	\bar{V}_j	—
log	$[1, 1] \odot \bar{V}_k$	—

$$\bar{D}_i^j \ni d_i^j \quad (3.2.3)$$

が成立する ($j = 1, \dots, r; i = 1, 2$) (表 1; $\oplus, \ominus, \otimes, \odot$ など) はそれぞれ機械区間演算の加算, 減算, 乗算, 除算を表す). 一方, それぞれの浮動小数点方式に応じて, 我々は発生誤差 δ_j を真を含む区間 $\bar{\Delta}_j$ を中間結果 \bar{v}_j 等に依存して決めることができる. すなわち, マシンエプシロン ε_M を用いると, 「計算値 \bar{v}_j が観察された場合の発生誤差の絶対値は $|\bar{v}_j| \cdot \varepsilon_M$ より小さい」ということが成立するから, $\bar{\delta}_j \equiv |\bar{v}_j| \cdot \varepsilon_M$ として[†], $\bar{\Delta}_j \equiv [-\bar{\delta}_j, \bar{\delta}_j]$ とすればよい. 以上より, (2.2.5) は

$$\begin{aligned} \bar{f} - f \in & (\bar{D}_1^1 \otimes [\bar{u}_{r1} - u_{r1}, \bar{u}_{r1} - u_{r1}] \\ & \oplus \bar{D}_2^2 \otimes [\bar{u}_{r2} - u_{r2}, \bar{u}_{r2} - u_{r2}]) \oplus \bar{\Delta}_r \end{aligned} \quad (3.2.4)$$

となる. $\bar{u}_{r1} - u_{r1}, \bar{u}_{r2} - u_{r2}$ についても同様に展開していくと, (2.2.7) は

$$\begin{aligned} \bar{f} - f \in & ((\dots (\bar{W}_r \otimes \bar{\Delta}_r \oplus \bar{W}_{r-1} \otimes \bar{\Delta}_{r-1}) \oplus \dots \\ & \dots) \oplus \bar{W}_2 \otimes \bar{\Delta}_2) \oplus \bar{W}_1 \otimes \bar{\Delta}_1 \end{aligned} \quad (3.2.5)$$

となる. \bar{W}_j ($j = 1, \dots, r$) は (2.2.6) の w_j に対応する機械区間であり, その計算は以下のように高速自動微分法を機械区間演算化することによって遂行される^{4), 5), 6), 8)}.

1) 初期化

$$\begin{aligned} \bar{W}_1, \dots, \bar{W}_{r-1} & := [0, 0]; \\ \bar{W}_r & := [1, 1]. \end{aligned}$$

2) 計算

for $j := r$ downto 1 do

$$\begin{cases} a := \psi_j \text{ の第 1 実引数の中間変数の添字番号;} \\ b := \psi_j \text{ の第 2 実引数の中間変数の添字番号;} \\ \bar{W}_a := \bar{W}_a \oplus \bar{W}_j \otimes \bar{D}_1^j; \\ \bar{W}_b := \bar{W}_b \oplus \bar{W}_j \otimes \bar{D}_2^j. \end{cases}$$

(ただし, ψ_j の仮引数が 1 個の場合や ψ_j の実引数が中間変数でない場合には, a または b のどちらか一方だけが定義される. その場合は \bar{W}_a や \bar{W}_b に関する計算の部分を削除する. \bar{F} が計算できるならば, 0 を含んだ機械区間による除算などが発生することなく \bar{W}_j の計算ができる. もちろん, \bar{D}_i^j を前もって計算する代わりに, 直接 $\bar{W}_j \otimes \bar{D}_i^j$ などの計算を行うこともできる⁴⁾.)

ここで重要なことは, 上の方法により r に比例した手間で, $\bar{W}_1, \dots, \bar{W}_r$ が計算できることである.

以上により,

$$A_F \equiv \bigoplus_{j=1}^r \bar{W}_j \otimes \bar{\Delta}_j$$

とおけば, $|\bar{f} - f| \leq A_F$ が厳密に成立する (\oplus は機械区間演算による総和).

[†] アンダーフローの場合も考慮すると, $pmin, nmax$ をそれぞれ機械で表現可能な正の最小浮動小数点数, 負の最大浮動小数点数として, $\bar{\delta}_j \equiv \max\{|\bar{v}_j| \cdot \varepsilon_M, pmin, -nmax\}$ とすることになる.

3.3. A_F の計算方法と計算の手間

上に述べてきた浮動小数点数の入力 $\bar{x}_1, \dots, \bar{x}_n$ に対する関数値 \bar{f} に含まれる丸め誤差の絶対値の厳密な上界, A_F , の計算方法を以下にまとめる:

- (1) f を計算する手続きに現れる演算を機械区間演算に置き換え, 変数を機械区間を記憶できるものに置き換える;
- (2) (1) で作成した手続きに $\bar{X}_1 = [\bar{x}_1, \bar{x}_1], \dots, \bar{X}_n = [\bar{x}_n, \bar{x}_n]$ を入力機械区間として与え, それにしたがって $\bar{V}_1, \dots, \bar{V}_r$ ($= \bar{F}$) の値を計算し, 計算過程を得る;
- (3) 要素的偏導関数計算と高速自動微分法の算法を機械区間演算によって実行し, $\bar{W}_1, \dots, \bar{W}_r$ を計算する;
- (4) $\bar{\Delta}_j \equiv [-\bar{\delta}_j, \bar{\delta}_j]$ ($\bar{\delta}_j \equiv |\bar{V}_j| \cdot \varepsilon_M$) とする ($j = 1, \dots, r$);
- (5) $S \equiv \bigoplus_{j=1}^r \bar{W}_j \otimes \bar{\Delta}_j$ とする;
- (6) $A_F \equiv |S|$.

機械区間演算の手間は, 対応する通常の実数演算の定数倍の手間である. (1), (2) の計算は, 通常機械区間演算を行っているだけなので, 関数計算の定数倍の手間で遂行される. 上の (3) の計算に必要な手間は, 高速自動微分法によれば, 関数計算の手間の定数倍である. したがって, A_F の計算に必要な手間は, 関数 f を計算するのに必要な手間の定数倍である. また, A_F の計算に必要な領域は, 高速自動微分法を用いるために, 関数 f を計算するのに必要な手間の定数倍となる.

3.4. 最適性

入力 x_1, \dots, x_n を与えて f を計算したときの丸め誤差を $R = |\bar{f} - f|$ と記すと,

$$R = \left| \sum_{j=1}^r w_j \cdot \delta_j \right| \quad (3.4.1)$$

である. 第 1 節の我々の仮定のもとでは, たまたま, $w_1 \cdot \delta_1, \dots, w_r \cdot \delta_r$ の符号がすべて一致する場合もありうる. その場合には各発生誤差は打ち消し合うことなく累積されて

$$R = \sum_{j=1}^r |w_j| \cdot |\delta_j| \quad (3.4.2)$$

となる. また, 発生誤差 δ_j の絶対値の上限の見積りを $\bar{\delta}_j$ ($|\delta_j| \leq \bar{\delta}_j$) とすれば, 最悪の場合, R の最大値

$$\bar{R} = \sum_{j=1}^r |w_j| \cdot \bar{\delta}_j \quad (3.4.3)$$

が実現される場合がありうる. 我々は \bar{R} を真を含む機械区間として A_F を得たが, さらに

$$\partial f / \partial v_j \neq 0 \quad (j = 1, \dots, r) \quad (3.4.4)$$

であると仮定すると, A_F が, 漸近的には (すなわち, マシンエプシロンが十分小さいときには), \bar{R} に一致するというを以下に示す.

定理

$$A_F \equiv \left| \bigoplus_{j=1}^r \bar{W}_j \otimes \bar{\Delta}_j \right|, \quad (3.4.5)$$

$$\bar{R} \equiv \sum_{j=1}^r |w_j| \cdot \bar{\delta}_j, \quad \bar{\Delta}_j \equiv [-\bar{\delta}_j, \bar{\delta}_j]$$

とおくと、任意の $\varepsilon (> 0)$ に対してある $\eta (> 0)$ が存在して、 η 以下のマシンエプシロン ε_M を持つ浮動小数点方式で計算すれば

$$1 \leq \frac{A_F}{\bar{R}} \leq 1 + \varepsilon \quad (3.4.6)$$

である。□

証明

まず、機械区間演算に関する定理[†] 1) から、次の

(i) が言える。

(ii) 任意の $\varepsilon' (> 0)$ に対してある $\eta_1 (> 0)$ が存在して、 η_1 以下の任意のマシンエプシロン ε_M を持つ浮動小数点方式において、(2.2.6) および §3.2 で定義した w_j, \bar{W}_j が

$$0 \leq |\bar{W}_j| - |w_j| \leq |\bar{W}_j - w_j| \leq \varepsilon' \quad (j = 1, \dots, r) \quad (3.4.7)$$

を満たす。

また、(3.4.4) と基本演算の連続微分可能性と \bar{F} の計算可能性の仮定とより、

(ii) $m \equiv \frac{1}{2} \min_j \left| \frac{\partial f}{\partial w_j} \right| (> 0)$ が存在し、 m に関して $\eta_2 (> 0)$ が存在して、 η_2 以下の任意のマシンエプシロン ε_M を持つ浮動小数点方式において計算すると、 $|w_j| > m$ が満たされる。

(i), (ii) より、

(iii) $\eta_3 \equiv \min\{\eta_1, \eta_2\}$ より小さいマシンエプシロン ε_M を持つ浮動小数点方式において計算した w_j, \bar{W}_j に対して、

$$1 \leq \frac{|\bar{W}_j|}{|w_j|} \leq 1 + \frac{\varepsilon'}{m} \quad (3.4.8)$$

となる。

$A_F/\bar{R} \geq 1$ であることは、 A_F の定義より明らか。一方、§3.3 の (5) における A_F の機械区間演算による内積計算で発生する丸め誤差まで考慮に入れても

$$A_F \leq \left(\sum_{j=1}^r |\bar{W}_j| \cdot \bar{\delta}_j \right) \cdot (1 + \varepsilon_M)^r \quad (3.4.9)$$

という不等式は成立する。これは、 $\bar{W}_j \otimes \bar{\Delta}_j$ の絶対値の上界は $|\bar{W}_j| \cdot \bar{\delta}_j \cdot (1 + \varepsilon_M)$ で抑えられること、加算を1回行う度に中間結果の上界が $1 + \varepsilon_M$ 倍されること、そして、(3.2.5) の内積計算には加算が $r - 1$ 回あることによる。

正の任意の $\varepsilon (< 1)$ に対して $\varepsilon' = \varepsilon \cdot m/2$ とし、(iii) により η_3 を決めることができる。 $\eta_4 \equiv \min\{\eta_3, \varepsilon/4r\}$ 以下のマシンエプシロン ε_M を持つ浮動小数点方式によって A_F を計算すれば、

$$(1 + x)^r \leq e^{rx} \leq 1 + e^{rx} \cdot r \cdot x \quad (x \geq 0, r > 0)$$

より、

$$\begin{aligned} \frac{A_F}{\bar{R}} &\leq \frac{\sum_{j=1}^r |\bar{W}_j| \cdot \bar{\delta}_j \cdot (1 + \varepsilon_M)^r}{\sum_{j=1}^r |w_j| \cdot \bar{\delta}_j} \\ &\leq \left(1 + \frac{\varepsilon'}{m}\right) \cdot (1 + e^{r\varepsilon_M} \cdot r \cdot \varepsilon_M) \\ &\leq 1 + \frac{\varepsilon'}{m} + 1.5 \cdot e^{1/4} \cdot \frac{1}{4} \cdot \varepsilon \\ &\leq 1 + \varepsilon. \end{aligned} \quad (3.4.10)$$

□

4. 数値実験

4.1. 機械区間演算の実現方法

我々は、Pascal-SC のような本格的な機械区間演算のためのソフトウェア²⁾を持っていないので、以下のようにして機械区間演算を模倣した。(我々は東京大学大型計算機センターの副システム VAX8600 を使用し、KCL(Kyoto Common Lisp) を用いてプログラミングを行った。なお、浮動小数点方式は2進(絶対値+符号)表現、丸めは0捨1入、マシンエプシロン ε_{M0} は 2^{-56} である。)

56桁2進表現を基礎に、56桁以下の任意の桁数の仮数部を持つ2進数の浮動小数点演算を実現し、それをマシンエプシロン ε_M で区別する。 ε_M で代表される浮動小数点方式における機械区間演算は、まず、

(1) 通常の区間演算の定義をそのまま ε_M の浮動小数点方式で計算し、

次に、

(2) (1) の結果の区間 $C = [c^l, c^h]$ の両端を広げて真の結果を含むような区間 $\bar{C} = [\bar{c}^l, \bar{c}^h]$ を定める; ここで、

$$\begin{aligned} \bar{c}^l &= \min\{c^l(1 - \varepsilon_M), c^l(1 + \varepsilon_M), \\ &\quad c^h(1 - \varepsilon_M), c^h(1 + \varepsilon_M)\}, \\ \bar{c}^h &= \max\{c^l(1 - \varepsilon_M), c^l(1 + \varepsilon_M), \\ &\quad c^h(1 - \varepsilon_M), c^h(1 + \varepsilon_M)\} \end{aligned}$$

である。[‡]

4.2. 例1: Ebers-Moll モデル

PNP トランジスタの Ebers-Moll モデル (図3) のベース電流の計算式を関数と見なした場合において、線形近似による丸め誤差の絶対評価 A_f 、第3節の A_F 、機械区間演算の結果 \bar{F} を、真の値を含む区間の幅 (それぞれ、 $2 \cdot A_f$ (近似値)、 $2 \cdot A_F$ 、 $d(\bar{F})$) について比較する。§4.1 の機械区間演算の模倣の ε_M を 2^{-12} 、 2^{-24} 、 2^{-36} 、 2^{-48} と変えて、図4に示す。この結果から判ることは、この程度の小規模な関数では、 $2 \cdot A_f$ 、 $2 \cdot A_F$ 、 $d(\bar{F})$ はみなほぼ同じ大きさであること; したがって、 \bar{F} を計算した後に A_F を計算しても実用的にみてあまり利点はないことである。

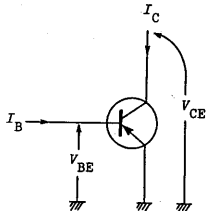
[†] 参考文献 1) の第4章の定理4および定理5。

[‡] 簡単のためアンダーフローの影響は無視した。

4.3. 例2: LU分解による線形方程式系の解

線形方程式系 $Ax = b$ の解 x は, LU分解などによって, 係数行列 A とベクトル b とから計算できる. したがって, この x の第1成分 x_1 を, A と b の関数 $f(A, b)$ とみなすことができる. ここでは, 入力 A, b からLU分解および前進, 後退代入により x_1 を計算するプログラムを計算の手続きとする関数 $f(A, b) = (1, 0, \dots, 0) \cdot A^{-1}b$ について, 実験を行う. まず, $[-1, 1]$ 上の一様分布に従う乱数を要素とする5次の行列とベクトルの対を10組用意する: $(A_1, b_1), \dots, (A_{10}, b_{10})$. $x_{1i} = f(A_i, b_i)$ に含まれる丸め誤差の推定値 $A_f, A_F, d(\bar{F})$ を, §4.2と同様に機械区間演算 (の模倣) の ϵ_M を $2^{-12}, 2^{-24}, 2^{-36}, 2^{-48}$ と変えて, 計算してある ($i = 1, \dots, 10$).

ϵ_M が大きい場合に, \bar{F} を計算することができない (したがって, A_F も計算できない) 場合がある. そのため, 線形近似による絶対評価 A_f を比較の基準とし, $A_F/A_f, d(\bar{F})/(2 \cdot A_f)$ を図5に示す. この図中で, ∞ の位置にあるものは, 0を含んだ区間による除算が発生したために \bar{F} の計算が中断されたものである. また, 10次の行列, ベクトルについても同様



$$I_B = -(1 - \alpha_F) \cdot I_{ES} \cdot [\exp(-q \cdot V_{BE}/k \cdot T) - 1] - (1 - \alpha_R) \cdot I_{CS} \cdot [\exp(q \cdot (V_{CE} - V_{BE})/k \cdot T) - 1]$$

$$I_C = -\alpha_F \cdot I_{ES} \cdot [\exp(-q \cdot V_{BE}/k \cdot T) - 1] + I_{CS} \cdot [\exp(q \cdot (V_{CE} - V_{BE})/k \cdot T) - 1]$$

- I_B, I_C : ベース電流, コレクタ電流
- I_{ES}, I_{CS} : エミッタ・ベース間飽和電流, コレクタ・ベース間飽和電流
- α_F, α_R : 電流伝達率
- V_{BE}, V_{CE} : ベース・エミッタ電圧, コレクタ・エミッタ電圧
- T : 温度
- q : 電子の電荷
- k : ボルツマン定数

図3. PNP トランジスタの Ebers-Moll モデル

の実験を行った (図6). 10次の行列での結果の一部を表2に示すが, A_F の保証する区間の幅に比べて \bar{F} の保証する区間の幅が桁違いに広いことがわかる.

以上のように, 計算過程が長くなること, 単なる機械区間演算の結果は我々の提案する推定値に比べてはなはだしい過大評価となることがある. たとえば, 線形方程式系の解については, 5次の場合で $10^1 \sim 10^5$ 倍, 10次の場合では $10^4 \sim 10^8$ 倍の過大な値を与える場合があるということが見られる. つまり, 我々の提案する方法は厳密な上界でありながら良い推定値を与えるということである. また, 通常の場合 (5次の場合 $\epsilon_M < 2^{-24}$, 10次の場合 $\epsilon_M < 2^{-48}$) では, この程度の問題に対しては線形近似 A_f も十分に良い評価値を与えていることもわかる.

最近には内積演算を正確に実行するようなハードウェアやソフトウェア²⁾が存在するようであるが, 上では乗算, 加算を行う度に個別に丸め誤差が発生する状況を考えた.

5. あとがき

我々は実際の計算機上で, 丸め誤差の絶対値の厳密な上界を与える算法を示した. しかし, その算法は通常の区間演算の与える結果に比べてずっと現実的な

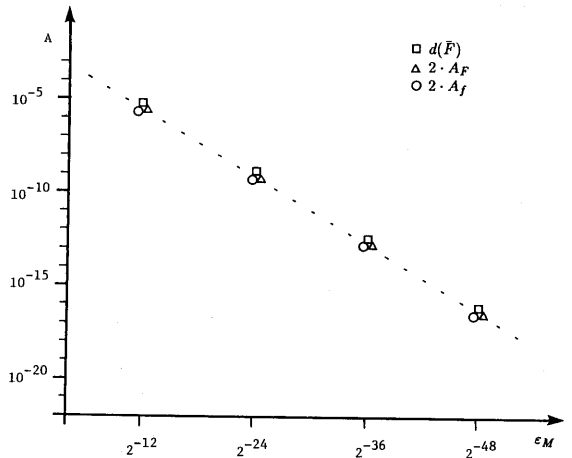


図4. Ebers-Moll モデルのベース電流の保証区間幅

$$V_{BE} = -0.4V, V_{CE} = -1.0V, I_{ES} = 1.0 \cdot 10^{-9}A,$$

$$I_{CS} = 2.0 \cdot 10^{-9}A, \alpha_F = 0.98, \alpha_R = 0.5, T = 300K,$$

$$q = 1.602 \cdot 10^{-19}C, k = 1.38066 \cdot 10^{-23}J/K, I_B \approx -1.04 \cdot 10^{-4}A$$

表2. 10元線形方程式系の解の第1成分の保証区間とマシンエプシロンの関係

ϵ_M	A_f から推定される 計算値の有効桁数	\bar{F} による 保証区間	A_F による 保証区間
2^{-12}	0.7438964843	—	—
2^{-24}	0.7542193532	[-3344.4582519, 3346.1162109]	[0.4637820125, 1.0446566939]
2^{-36}	0.7542197855	[0.0575954438, 1.4508441332]	[0.7542197555, 0.7542198151]
2^{-48}	0.7542197855	[0.7540463030, 0.7543932679]	[0.7542197855, 0.7542197855] [†]

[†][0.754219785475757, 0.7542197854840781]

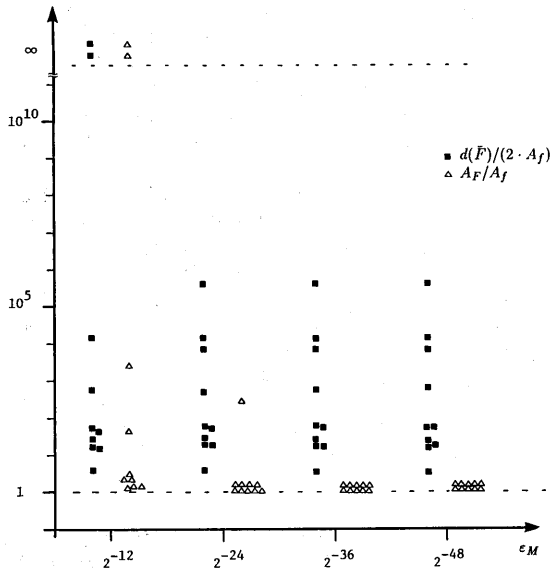


図5. 5元線形方程式系の解の第1成分の保証区間幅の比較

(“∞”の記号のところにある点は零を含む区間による除算が発生したために区間演算が中断されたものを示す.)

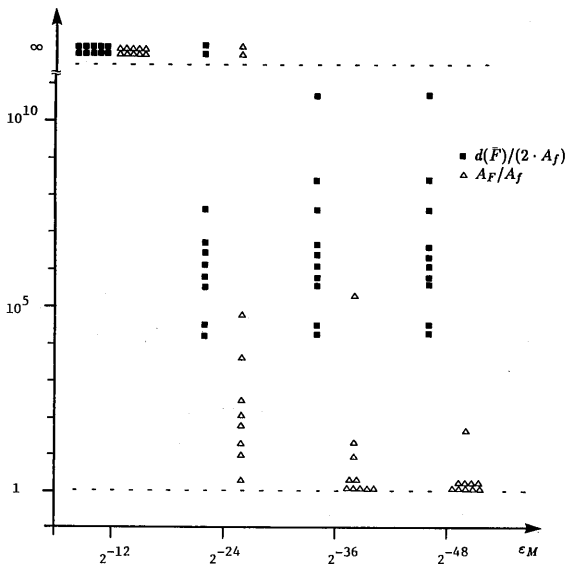


図6. 10元線形方程式系の解の第1成分の保証区間幅の比較

(“∞”の記号のところにある点は零を含む区間による除算が発生したために区間演算が中断されたものを示す.)

上界の評価値を与え、計算の時間からみても十分に実用的であること、さらに、適当な条件のもとでは、その算法が与える上界がマシンエプシロンが小さいときに真の丸め誤差の絶対値の上限に漸近することを証明した。

通常は丸め誤差評価は線形近似でも十分であるが、計算で得られた値と真の値との喰い違いを厳密かつ最適に評価することは、計算結果の品質を保证するという観点から重要なことである。また、その際、高速自動微分法の考え方が本質的に有効な役割を果していることにも注目すべきであろう。

参考文献

- 1) G. Alefeld and J. Herzberger: *Introduction to Interval Computations*. Academic Press, New York, 1983.
- 2) G. Bohlander, C. Ullrich, J. W. Gudenberg, and L. B. Rall: *Pascal-SC — A Computer Language for Scientific Computation*. Academic Press, Orlando, 1987.
- 3) E. R. Hansen: A Generalized Interval Arithmetic. K. Nickel (ed.): *Interval Mathematics*, Lecture Notes in Computer Science 29, Springer-Verlag, Berlin, 1975, pp. 7–18.
- 4) M. Iri: Simultaneous Computation of Functions, Partial Derivatives and Estimates of Rounding Errors — Complexity and Practicality. *Japan Journal of Applied Mathematics*, Vol. 1, No. 2 (1984), pp. 223–252.
- 5) M. Iri and K. Kubota: Methods of Fast Automatic Differentiation and Applications. *Research Memorandum RMI 87-02*, Department of Mathematical Engineering and Information Physics, University of Tokyo, 1987.
- 6) M. Iri, T. Tsuchiya, and M. Hoshi: Automatic Computation of Partial Derivatives and Rounding Error Estimates with Applications to Large-scale Systems of Nonlinear Equations. To appear in *Journal of Computational and Applied Mathematics*. (伊理正夫, 土谷隆, 星守: 偏導関数計算と丸め誤差推定の自動化の大規模非線形方程式系への応用. *情報処理*, Vol. 26, No. 11 (1985), pp. 1411–1420. の改定版)
- 7) G. Kedem: Automatic Differentiation of Computer Programs. *ACM Transactions on Mathematical Software*, Vol. 6, No. 2 (1980), pp. 150–165.
- 8) 久保田光一, 伊理正夫: 高速自動微分法の定式化と計算複雑度の解析. *情報処理学会論文誌*, Vol. 29, No. 6 (1988), pp. 551–560.
- 9) Yu. V. Matiyasevich: Veshchestvennye Chisla i ÈVM. *Kibernetika i Vychislitel'naya Tekhnika*, Vypusk 2 (1986), pp. 104–133.
- 10) L. B. Rall: Improved Bounds for Ranges of Functions. K. Nickel (ed.): *Interval Mathematics 1985*, Lecture Notes in Computer Science 212, Springer-Verlag, Berlin, 1985, pp. 143–155.