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†††

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示す．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”とを組み合わせたものである．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上限の良い近似値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．

1. まえがき

高速自動微分法[4)-6),8)]を用いることによって，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．すでに，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限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（関数）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，実際的な方法がなかった．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，本論文では，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ないと仮定し，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わかり，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．

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による丸めのない区間解析の方法[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らによって述べられていたが[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章で，取り扱う関数の範囲と丸め誤差を伴う計算について説明し，第3章では機械区間演算，算法，最適性等について述べる．第4章では，数値実験結果について述べる．

† Estimates of Rounding Errors with Fast Automatic Differentiation and Interval Analysis by KOICHI KUBOTA (Department of Administration Engineering, Faculty of Science and Technology, Keio University) and MASAO IRI (Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo).

†† 慶応義塾大学理工学部管理工学科
††† 東京大学工学部計数工学科

2. 丸め誤差を伴う計算

2.1 分解可能関数と計算過程[4)~6),8)]

関数を計算する過程で使用することのできる演算を+, −, ×, /, 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良い）．入力の値を定めて関数の手続きに従って計算すれば，関数値が得られる．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 $f(x_1, \cdots, x_n)$ 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従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，誤差のない計算（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かか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．）

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

$v_1, \cdots, v_r$：中間変数；$\psi_1, \cdots, \psi_r$：基本演算；
u_{j1}, u_{j2}：ψ_j の引数（入力変数 $x_1, \cdots, x_n$，定数，中間変数 $v_1, \cdots, v_{j-1}$ のいずれかがこれらに対する実引数となる（$j=1, \cdots, r$））．

図1 計算過程
Fig. 1 Computational process

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—ということを意味する．それゆえ，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．実際には誤差のない計算の遂行は不可能であるので，誤差のない計算過程の構造を直接知ることはできないが，第3章で述べるように，区間解析の手法を用いることにより，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，

$$v_1 \leftarrow \bar{\psi}_1(\bar{u}_{11}, \bar{u}_{12}) = \psi_1(\bar{u}_{11}, \bar{u}_{12}) + \delta_1$$
$$\vdots$$
$$v_j \leftarrow \bar{\psi}_j(\bar{u}_{j1}, \bar{u}_{j2}) = \psi_j(\bar{u}_{j1}, \bar{u}_{j2}) + \delta_j$$
$$\vdots$$
$$(\bar{f} =) v_r \leftarrow \bar{\psi}_r(\bar{u}_{r1}, \bar{u}_{r2}) = \psi_r(\bar{u}_{r1}, \bar{u}_{r2}) + \delta_r$$

$v_1, \cdots, v_r$: 中間変数; $\bar{\psi}_1, \cdots, \bar{\psi}_r$: 基本演算;
$\bar{u}_{j1}, \bar{u}_{j2}$: $\bar{\psi}_j$ の引数 (入力変数 $x_1, \cdots, x_n$, 定数, 中間変数 $v_1, \cdots, v_{j-1}$ のいずれかがこれらに対する実引数となる ($j=1, \cdots, r$));
$\delta_1, \cdots, \delta_r$: 発生誤差.

図 2 丸めつき計算過程
Fig. 2 Computational process with rounding errors.

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

によって定義する (図 2). (2.2.1)式の右辺 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

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

である ($0 < {}^{\exists}\theta_j < 1$). $\partial\psi_j/\partial u_{j1}$, $\partial\psi_j/\partial u_{j2}$ 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)~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

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

と記す. これを用いると, (2.2.2)式は

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

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:

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

次に, w_j ($j = 1, \cdots, r$) なる量を定義する. 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. ここで, 頂点 $v \in V$ は, 中間変数 (または, 入力変数, 定数) に対応し, 枝 $e \in E$ 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) $e_1, \cdots, e_l (\partial^+ e_1 = v_j,\ \partial^- e_k = \partial^+ e_{k+1}\ (k = 1, \cdots, l-1),\ \partial^- e_l = v_m)$ 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, 各有向道に対応する $\{(s_k, i_k)\}_{k=1}^{l}$ という列の集合を Q_j と記す. これらの諸量を用いて w_j を次のように定義する:

$$w_j \equiv \sum_{\{(s_k, i_k)\}_{k=1}^{l} \in Q_j} d_{i_l}{}^{s_l} \cdot d_{i_{l-1}}{}^{s_{l-1}} \cdots d_{i_1}{}^{s_1}$$
$$\left(= \sum_{\{(s_k, i_k)\}_{k=1}^{l} \in Q_j} \frac{\partial \psi_r}{\partial u_{r i_l}} \cdot \frac{\partial \psi_{s_{l-1}}}{\partial u_{s_{l-1} i_{l-1}}} \cdots \frac{\partial \psi_{s_1}}{\partial u_{s_1 i_1}} \right) \quad (j = 1, \cdots, r-1). \qquad (2.2.6)$$

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

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

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

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

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

である. 「絶対評価」, 「確率評価」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 にかかわらず一定とみなすものである:

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

ここで, $\partial f/\partial v_j$ ($j = 1, \cdots, r$) は高速自動微分法により, 実用的な手間で計算できる (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 $|\delta_j| \le |v_j| \cdot \varepsilon_M$ (ε_M: マシンエプシロン) が成立することから,

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

* 文献 5), 8) において, ω は各頂点に基本演算を対応させる関数, n は各枝が基本演算の何番目の引数であるかを表す関数, d は各枝に要素的偏導関数 (丸め誤差なし) を対応づける関数であると定義した. 本稿の $d_i{}^j$ は対応する $d(e)$ の計算誤差まで考慮に入れた値である.

と表される絶対評価 A_f は，丸め誤差の絶対値の上限の実用的には十分に良い近似であるが[4)~6),8)]，線形近似 L_f がすでに丸め誤差の近似であるので，厳密な上界とは言えない．

3. 丸め誤差の厳密な上界を推定する算法

$\bar{f}-f\in S$ なる区間* $S=[s^l, s^h]$ を計算する算法を以下に記す．丸め誤差の絶対値の厳密な上界は $|\bar{f}-f|\leq|S|$ によって与えられる．(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！）に現れるすべ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 $[\bar{x}_1, \bar{x}_1]$, …, $[\bar{x}_n, \bar{x}_n]$ 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．($\bar{F}$ 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 width$(\bar{F})$ も丸め誤差の厳密な上界であるが，f の計算ステップの数が増えると $\bar{F}$ の幅は一般にははなは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数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節で述べた仮定—関数値計算の実行可能性—の成立が保証される．なぜならば，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—ということは，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を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，

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

であり，これらが引数として現れるときには，

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

という形をとる（ここで，$\bar{U}_{ji}$ は引数 u_{ji} に対応する機械区間を示す）．

つぎに，要素的偏導関数 $\partial\phi_j/\partial u_{ji}$ 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，$0\leq\theta_j\leq1$ であることから，

$$\bar{D}_i{}^j\ni d_i{}^j \tag{3.2.3}$$

が成立する ($j=1,\cdots,r$; $i=1,2$) (**表 1**; ⊕, ⊖, ⊗, ⊘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，発生誤差 δ_j を真に含む区間 $\bar{\Delta}_j$ 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，「計算値が $\bar{v}_j$ であるとき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*，$\bar{\Delta}_j\equiv[-\bar{\delta}_j, \bar{\delta}_j]$ とすれば良い．以上より，(2.2.5)式は

$$\bar{f}-f\in(\bar{D}_1{}^r\otimes[\bar{u}_{r1}-u_{r1}, \bar{u}_{r1}-u_{r1}] \oplus\bar{D}_2{}^r\otimes[\bar{u}_{r2}-u_{r2}, \bar{u}_{r2}-u_{r2}])\oplus\bar{\Delta}_r \tag{3.2.4}$$

---

* 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．

* アンダ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．

表 1　要素的偏導関数に対応する機械区間
Table 1　Machine intervals corresponding to elementary partial derivatives.

$\bar{D}_i^j$ は $\partial\psi_j/\partial u_{ji}$ に対応する．ただし，$v_j=\psi_j(u_{j1},u_{j2})\Big|_{u_{j1}=v_k,\,u_{j2}=v_l}$．

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

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)式は

$$\bar{f}-f\in((\cdots(\overline{W}_r\otimes\bar{\Delta}_r\oplus\overline{W}_{r-1}\otimes\bar{\Delta}_{r-1})\oplus\cdots)\oplus\overline{W}_2\otimes\bar{\Delta}_2)\oplus\overline{W}_1\otimes\bar{\Delta}_1 \qquad (3.2.5)$$

となる．$\overline{W}_j$ $(j=1,\cdots,r)$ は (2.2.6)式の 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)~6),8)]．

1) 初　期　化

$\overline{W}_1,\cdots,\overline{W}_{r-1}:=[0,\ 0]$;

$\overline{W}_r:=[1,\ 1]$.

2) 計　　算

for $j:=r$ **downto** 1 **do**

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

(ただし，ϕ_j の仮引数が1個の場合や ϕ_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 $\overline{W}_a$ や $\overline{W}_b$ 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 $\overline{W}_j$ 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 $\overline{W}_j\otimes\bar{D}_1^j$ 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，$\overline{W}_1,\cdots,\overline{W}_r$ が計算できることである．

以上により，

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

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

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 $\bar{x}_1,\cdots,\bar{x}_n$ 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 $\bar{X}_1=[\bar{x}_1,\bar{x}_1],\cdots,\bar{X}_n=[\bar{x}_n,\bar{x}_n]$ を入力機械区間として与え，それに従って $\bar{V}_1,\cdots,\bar{V}_r(=\bar{F})$ 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，$\overline{W}_1,\cdots,\overline{W}_r$ 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 $(j=1,\cdots,r)$；

(5)　$S\equiv\bigoplus_{j=1}^{r}\overline{W}_j\otimes\bar{\Delta}_j$ 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，通常の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（**総演算回数**）の定数倍となる．

3.4　最　適　性

入力 $x_1,\cdots,x_n$ 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，

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

である．第1章の我々の仮定のもとでは，たまたま，$w_1\cdot\delta_1,\cdots,w_r\cdot\delta_r$ 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

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

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

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

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

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

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)，$\bar{R}$ に一致するということを以下に示す．

定理

基本演算の連続微分可能性，$\bar{F}$の計算可能性および(3.4.4)式の成立を仮定する．上述の記号を用いて

$$A_F \equiv \left| \bigoplus_{j=1}^{r} \overline{W}_j \otimes \bar{\Delta}_j \right|, \qquad \bar{R} \equiv \sum_{j=1}^{r} |w_j| \cdot \delta_j, \quad \bar{\Delta}_j \equiv [-\delta_j, \delta_j] \tag{3.4.5}$$

とおくと，任意の ε (>0) に対してある η (>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

$$1 \leq \frac{A_F}{\bar{R}} \leq 1+\varepsilon \tag{3.4.6}$$

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．

(i) 任意の ε' (>0) に対してある η_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)式および 3.2 節で定義した $w_j, \overline{W}_j$ が

$$0 \leq |\overline{W}_j| - |w_j| \leq |\overline{W}_j - w_j| \leq \varepsilon' \qquad (j=1, \cdots, r) \tag{3.4.7}$$

を満たす．

また，(3.4.4)式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により，

(ii) $m \equiv 1/2 \min_j |\partial f/\partial v_j|$ (>0) が存在し，m に関して η_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．

そこで，(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, $\overline{W}_j$ に対して

$$1 \leq \frac{|\overline{W}_j|}{|w_j|} \leq 1+\frac{\varepsilon'}{m} \tag{3.4.8}$$

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 節の(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

$$A_F \leq \left(\sum_{j=1}^{r} |\overline{W}_j| \cdot \delta_j \right) \cdot (1+\varepsilon_M)^r \tag{3.4.9}$$

という不等式は成立する．これは，$\overline{W}_j \otimes \bar{\Delta}_j$ の絶対値の上限は $|\overline{W}_j| \cdot \delta_j \cdot (1+\varepsilon_M)$ で抑えられること，加算を1回行う度に中間結果の上限が $1+\varepsilon_M$ 倍されること，そして，(3.2.5)式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 ε (<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，

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

より，

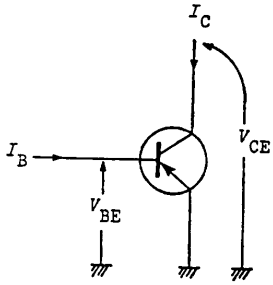
$$\begin{aligned} \frac{A_F}{\bar{R}} &\leq \frac{\left(\sum_{j=1}^{r} |\overline{W}_j| \cdot \delta_j \right) \cdot (1+\varepsilon_M)^r}{\sum_{j=1}^{r} |w_j| \cdot \delta_j} \\ &\leq \left(1+\frac{\varepsilon'}{m}\right) \cdot (1+e^{r\varepsilon_M} \cdot r \cdot \varepsilon_M) \\ &\leq 1+\frac{\varepsilon'}{m}+1.5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

□

4. 数値実験

4.1 機械区間演算の実現方法

Pascal-SC のような本格的な機械区間演算のための



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

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

図 3 PNP トランジスタの Ebers-Moll モデル
Fig. 3 Ebers-Moll model for a PNP-transistor.
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

* 文献 1)の第4章の定理4および定理5.

ソフトウェア[2]には頼らずに，以下のようにして機械区間演算を模倣した．(東京大学大型計算機センターの副システム VAX 8600 の上で KCL (Kyoto Common Lisp) を用いた．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，

$$\bar{c}^l=\min\{c^l(1-\varepsilon_M), c^l(1+\varepsilon_M), c^h(1-\varepsilon_M), c^h(1+\varepsilon_M)\},$$

$$\bar{c}^h=\max\{c^l(1-\varepsilon_M), c^l(1+\varepsilon_M), c^h(1-\varepsilon_M), c^h(1+\varepsilon_M)\}$$

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，線形近似による丸め誤差の絶対評価 A_f，第3章の 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)，$2\cdot A_F$，width $(\bar{F})$) について比較する．4.1 節の機械区間演算の模倣の ε_M を 2^{-12}, 2^{-24}, 2^{-36}, 2^{-48} と変えて計算した結果を図 4 に示す．この結果からわかることは，この程度の小規模な関数では，$2\cdot A_f$, $2\cdot A_F$, width $(\bar{F})$ 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．

* 簡単のためアンダフローの影響は無視した．

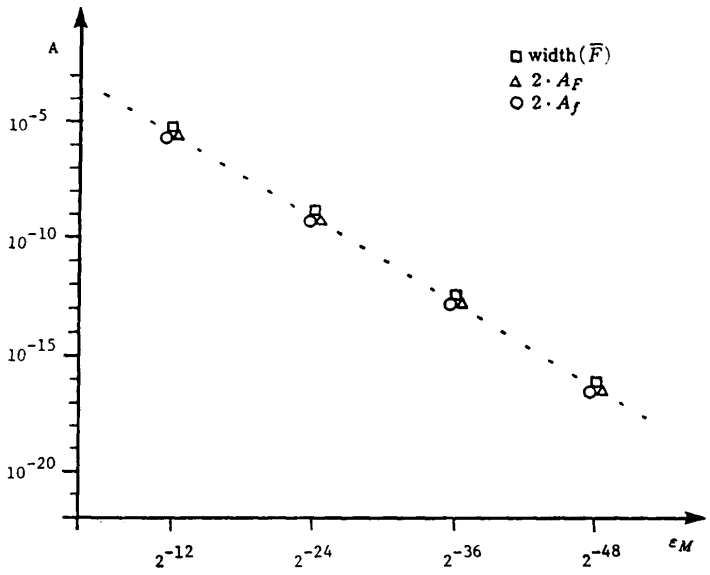


図 4 Ebers-Moll モデルのベース電流の保証区間幅

Fig. 4 Comparison of the widths of guaranteed intervals containing the exact value of the function which is the base current in the Ebers-Moll model where, $V_{BE}=-0.4$V, $V_{CE}=-1.0$V, $I_{ES}=1.0^{-9}$A, $I_{CS}=2.0^{-9}$A, $\alpha_F=0.98$, $\alpha_R=0.5$, $T=300$K, $q=1.602^{-19}$C, k$=1.38066^{-23}$J/K, $I_B\simeq-1.04\times10^{-4}$A

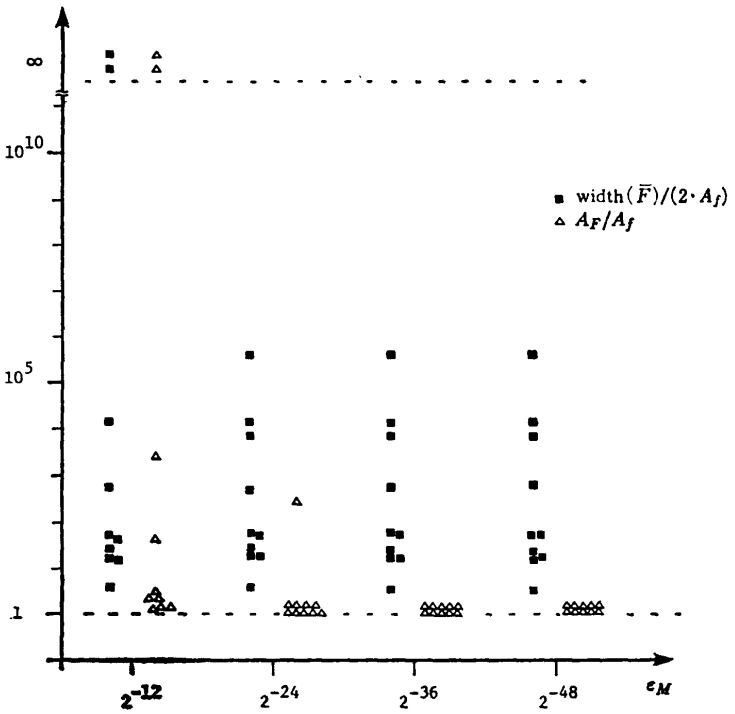


図 5 5元線形方程式系の解の第1成分の保証区間幅の比較

Fig. 5 Comparison of the widths of guaranteed intervals for 5-dimensional linear systems.
("∞" 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．)

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 分解法などによって，係数行列 A とベクトル b とから計算できる．そこで，この x の第 1 成分 x_1 を，A と b の関数 $f(A,b)$ と見なすことができる．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手続きとする関数 $f(A,b)=(1,0,\cdots,0)\cdot A^{-1}b$ 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), $\cdots$, (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, width $(\bar{F})$ を，4.2 節と同様に，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，計算する $(i=1,\cdots,10)$.

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, width$(\bar{F})/(2\cdot A_f)$ 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．

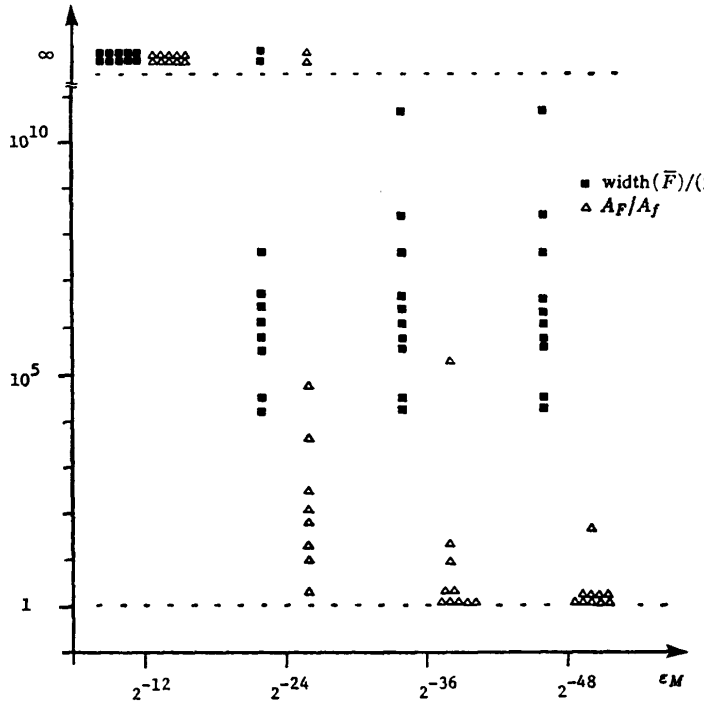


図 6 10 元線形方程式系の解の第 1 成分の保証区間幅の比較
Fig. 6 Comparison of the widths of guaranteed intervals for 10-dimensional linear systems.
（"∞" の記号のところにある点は零を含む区間による除算が発生したために区間演算が中断されたものを示す．）

以上のように，計算過程が長くなると，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 10^1〜10^5 倍，10 次の場合では 10^4〜10^8 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には厳密な上界でありながらかつ上限に近い値を与えるという特長がある．また，通常の状況（5 次の場合 $\varepsilon_M\leq 2^{-24}$, 10 次の場合 $\varepsilon_M\leq 2^{-48}$）では，この程度の問題に対しては線形近似 A_f も上限の十分に良い評価値を与えていることもわかる．

最近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

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との関係
Table 2 Guaranteed intervals with $\bar{F}$ and A_F for a 10-dimensional linear system.

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}	0.7542197853	[0.0575954438,1.4508441332]	[0.7542197555,0.7542198151]
2^{-48}	0.7542197855	[0.7540463030,0.7543932679]	[0.7542197855,0.7542197855]†

†[0.754219785475757,0.7542197854840781]

の絶対値の厳密な上界を与える算法を示した．しかも，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上界の評価値を与え，計算の手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．

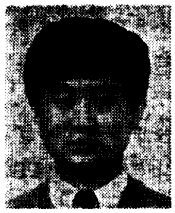
通常は丸め誤差評価は線形近似でも十分であるが，計算で得られた値と真の値との食い違いを厳密かつ最適に評価することは，計算結果の品質を保証するという観点から重要なことである．また，その際，高速自動微分法の考え方が本質的に有効な役割を果たしていることにも注目すべきであろう．

謝辞 本研究をまとめるにあたり有益な助言をくださった東京大学工学部計数工学科室田一雄助教授に感謝いたします．なお，本研究の一部は文部省科学研究費補助金（一般研究(B)63460131：高速自動微分法の実用化と応用分野の開拓）の援助によるものである．

参考文献

1) Alefeld, G. and Herzberger, J.: *Introduction to Interval Computations*, Academic Press, New York (1983).
2) Bohlender, G., Ullrich, C., Gudenberg, J. W. and Rall, L. B.: *Pascal-SC—A Computer Language for Scientific Computation*, Academic Press, Orlando (1987).
3) Hansen, E. R.: A Generalized Interval Arithmetic, in Nickel, K. (ed.), *Interval Mathematics*, Lecture Notes in Computer Science 29, pp. 7-18, Springer-Verlag, Berlin (1975).
4) Iri, M.: Simultaneous Computation of Functions, Partial Derivatives and Estimates of Rounding Errors—Complexity and Practicality, *Jpn. J. Appl. Math.*, Vol. 1, No. 2, pp. 223-252 (1984).
5) Iri, M. and Kubota, K.: Methods of Fast Automatic Differentiation and Applications, *Research Memorandum*, RMI 87-02, Department of Mathematical Engineering and Information Physics, University of Tokyo (1987).
6) Iri, M., Tsuchiya, T. and Hoshi, M.: Automatic Computation of Partial Derivatives and Rounding Error Estimates with Applications to Large-scale Systems of Nonlinear Equations, *J. Comput. Appl. Math.*, Vol. 24, No. 3, pp. 365-392 (1988). (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, pp. 1411-1420 (1985) の改定版.)
7) Kedem, G.: Automatic Differentiation of Computer Programs, *ACM Trans. Math. Softw.*, Vol. 6, No. 2, pp. 150-165 (1980).
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, pp. 551-560 (1988).
9) Matiyasevich, Yu. V.: Veshchestvennye Chisla i ÈVM, *Kibernetika i Vychislitel'naya Tekhnika*, Vypusk 2, pp. 104-133 (1986).
10) Rall, L. B.: Improved Bounds for Ranges of Functions, in Nickel, K. (ed.), *Interval Mathematics 1985*, Lecture Notes in Computer Science 212, pp. 143-155, Springer-Verlag, Berlin (1985).

(昭和63年12月1日受付)
(平成元年4月11日採録)



久保田光一（正会員）

1960年生．1983年東京大学工学部計数工学科卒業．1985年同大学大学院修士課程修了．現在慶応義塾大学理工学部管理工学科助手．数値計算の研究に従事．ACM，日本 OR 学会各会員．

伊理　正夫（正会員）

昭和8年生．昭和30年東京大学工学部応用物理学科(数理工学専修)卒業．昭和35年同大学院博士課程修了．工学博士．九州大学工学部助手，助教授（通信工学科），東京大学助教授（工学部計数工学科）を経て，現在同大教授．回路，グラフ，数値計算，言語などの研究，教育に従事．昭和40年松永賞受賞．著書「Network Flow, Transportation and Scheduling」など．