解説

アフィン演算について†

宮田 孝富*

1 はじめに

現在、非線形現象を含む回路や各種システムの解析は、計算機上の離散的かつ有限精度の計算に置きかえられて行われることが多い。その計算の過程で生じるさまざまな誤差を、計算機の浮動小数点演算を用いながら、数学的に正しく、かつ実用的なレベルで見積もるための方法を総称して「精度保証付き数値計算」と言う。

精度保証付きで非線形方程式などを解く場合、その理論的な根拠は関数解析の分野であられる不動点定理に依ることが多い。また、丸め誤差を含む計算機の浮動小数点演算を用いて、定理が成立しているかどうかを数値的に確かめる過程においては、関数の区間包囲を求めるうえでも計算機の丸め誤差を厳密に把握するうえでも区間演算の技法が不可欠である。

区間演算の欠点として、確かに計算された区間は真の値を含むものの、区間幅が極端に広がってしまうことが多い点が挙げられる。関数の値域を評価する場合のように、入力区間がある程度広い場合にはこの欠点がより深刻な形で現われやすい。その原因には2つあり、1つは「区間同士の加減算では打ち消しが生じないことによる過大評価」、もう1つは「wrapping effect」である。これらの問題を解決するための方法としては、平均値形式などがよく知られている。

本稿では、区間演算における区間幅の広がりを抑えるための方法の一つとして、変数間の相関性を考慮した特殊な区間演算方式であるアフィン演算を紹介する。

2 アフィン演算概説

アフィン演算は1994年に J.Stolfi ら [1] によって初めて提案され、1997年に Luiz H. de Figueiredo の LNCC(ブラジル国立科学計算研究所)学位論文 [2] によって体系的にまとめられた区間演算の一種である。アフィン形式という特殊な形式を用いることで、区間の上限・下限のみならず変数間相互の相関関係をも表現できる。そのため、アフィン演算においては加減算

で起きるべき打ち消しが生じるし、同類項もまとめられる。さらに wrapping effect の影響をまったく受けない。

2.1 通常の区間演算や平均値形式との比較

$x \in [0.01, 0.49]$ として、 $f(x) = x - 0.8\sqrt{x}$ の値を計算してみる。区間演算の定義に従って計算すると結果は

$$\begin{aligned} f([0.01, 0.49]) &= [0.01, 0.49] - 0.8\sqrt{[0.01, 0.49]} \\ &= [0.01, 0.49] - [0.08, 0.56] \\ &= [0.01 - 0.56, 0.49 - 0.08] \\ &= [-0.55, 0.41] \end{aligned} \quad (1)$$

となる。 x と $0.8\sqrt{x}$ とは連動して動き、例えば区間の下端に関して $x=0.01$ のときには $0.8\sqrt{x}=0.56$ とならないので、この計算では実際に取り得る値より無駄な(区間幅の広い)評価をしていることになる。

次に、この $f(x)$ を平均値形式によって計算してみる。平均値形式は $f(x)$, $f: \mathbf{R}^n \rightarrow \mathbf{R}^n$ に対して、 $\mathbf{R}^n$ 内の有界な閉区間 I の像 $f(I) = \{f(x) | x \in I\}$ を

$$f(c) + F'(I)(I - c) \quad (2)$$

で与える方法であった。ここで F' は f の導関数 f' の区間包囲である。また c は区間ベクトル I の中心点ベクトルとする。つまり今の場合、

$$I = [0.01, 0.49] \quad (3)$$

$$c = 0.25 \quad (4)$$

$$f(c) = -0.15 \quad (5)$$

$$\begin{aligned} F'(I) &= 1 - \frac{0.4}{\sqrt{[0.01, 0.49]}} \\ &= [-3, \frac{3}{7}] \end{aligned} \quad (6)$$

$$I - c = [-0.24, 0.24] \quad (7)$$

$$\begin{aligned} f(I) &= -0.15 + [-3, \frac{3}{7}] [-0.24, 0.24] \\ &= [-0.87, 0.57] \end{aligned} \quad (8)$$

このように、平均値形式は区間演算の定義に従って計算した場合より悪い結果を与えることもある。一般に

† On Affine Arithmetic
Takatomi MIYATA

* 金沢工業大学 工学部 情報工学科
Kanazawa Institute of Technology

$$\max_{x \in I} |F'(x)| \quad (9)$$

の値が大きいきは平均値形式による評価が有効であるとは言えない。

そこで x を区間 $[0.01, 0.49]$ で表すかわりに次のようにおいてみることにする：

$$x = 0.25 + 0.24\epsilon_1, (-1 \leq \epsilon_1 \leq 1) \quad (10)$$

ここで、 ϵ_1 は -1 から 1 までの値を取ることでしかわかっていないダミー変数である。また、 $0.01 \leq x \leq 0.49$ において常に

$$x + 0.07 \leq 0.8\sqrt{x} \leq x + 0.16 \quad (11)$$

であることから、同様に -1 から 1 まで動くダミー変数 ϵ_2 を新たに導入して

$$\begin{aligned} 0.8\sqrt{x} &= (x + 0.115) + 0.045\epsilon_2 \\ &= 0.365 + 0.24\epsilon_1 + 0.045\epsilon_2 \end{aligned} \quad (12)$$

とにおいてみる。その結果は次のようになる：

$$\begin{aligned} f(I) &= (0.25 + 0.24\epsilon_1) - (0.365 + 0.24\epsilon_1 + 0.045\epsilon_2) \\ &= -0.115 - 0.045\epsilon_2 \end{aligned} \quad (13)$$

式(13)は $-1 \leq \epsilon_2 \leq 1$ のもとで $[-0.16, -0.07]$ の間の値を取る。つまり、今の例に関する限り、この計算方法は前2者よりはるかにすぐれた評価を与える。

この計算の中でとくに注意すべき点は、 ϵ_1 が -1 から 1 のうちのどの値を取るかにかかわらず ϵ_1 の項を打ち消すことができる点である。言い換えれば、通常の区間演算の場合と異なり、この計算方法では減算時に (ϵ_1 の項の) 打ち消しがきちんと起きている。

ところで、今の例のような簡単な計算であれば、その正確な値域を下表1のような増減表を用いて容易に調べることができる(ただし、一般には導関数や極値を求めるのが難しいことも多い)。この結果から、ダミー変数 ϵ を導入する方法は今の例に関して正確な値域を与えたことがわかる。

表1 $f(x)$ の増減表(真の像は $[-0.16, -0.07]$)

x	0.01		0.16		0.49
$f'(x)$	-3	-	0	+	$\frac{3}{7}$
$f(x)$	-0.07	$\searrow$	-0.16	$\nearrow$	-0.07

では、次にもう1つの例として、以下のような x, A :

$$x = \begin{pmatrix} [-1, 1] \\ [-1, 1] \end{pmatrix}, A = \begin{pmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{pmatrix} \quad (14)$$

に対して積 Ax を考える。式(14)を区間演算の定義にしたがって計算すると最終的に、

$$Ax = \begin{pmatrix} [-\sqrt{2}, \sqrt{2}] \\ [-\sqrt{2}, \sqrt{2}] \end{pmatrix} \quad (15)$$

を得る。すなわち、真の像は明らかに x を原点中心に $\frac{\pi}{4}$ 回転させた正方形(図1参照、 Ax (exact) と名付けてある領域)であるのだが、区間演算による計算結果は式(15)のように、真の像を wrap する正方形(図1で Ax (IA) と名付けてある領域)となっている。

また、この結果にさらに A をかけて…ということを繰り返すと、真の像は同じ大きさの正方形がただ原点中心に $\frac{\pi}{4}$ ずつ回転し続けるだけなのに対して、区間演算で計算した結果は行列 A をかけるごとにどんどん大きな正方形になってしまう。

これは、一般にユークリッド空間内において、区間ベクトルでは軸に平行な超直方体領域しか表現できないために起こる現象で、wrapping effect と呼ばれる。

そこで、先ほどと同様に x を次のようにおいてみる：

$$x = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix}, (-1 \leq \epsilon_i \leq 1) \quad (16)$$

ここで、今の場合ベクトル x の成分間には相関がなく、各成分ごとに別のダミー変数を用いたことに注意する。このとき、積 Ax を計算した結果は次のようになる：

$$Ax = \begin{pmatrix} \frac{\sqrt{2}}{2} \epsilon_1 - \frac{\sqrt{2}}{2} \epsilon_2 \\ \frac{\sqrt{2}}{2} \epsilon_1 + \frac{\sqrt{2}}{2} \epsilon_2 \end{pmatrix} \quad (17)$$

式(17)が $-1 \leq \epsilon_1 \leq 1, -1 \leq \epsilon_2 \leq 1$ のもとでどのよう

図1 wrapping effect

な領域を動かすか考える。式(17)の右辺は

$$\begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \epsilon_1 + \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \epsilon_2 \quad (18)$$

のように書き直すことができるが、式(18)において第1項だけ考えた($\epsilon_2 = 0$ とした)ときは図2左に示すような線分上を動く。次に ϵ_2 も -1 から 1 まで動かすと図2右側のような領域を動く。

図2 式(17)の取りうる領域

これは真の像 Ax (exact) に一致する。言い換えれば、ダミー変数 ϵ を用いたこの計算方法では wrapping effect の影響を完全に排除している。

大変まわりくどい説明になったが、このダミー変数 ϵ を用いた方法こそが実はアフィン演算による方法なのである。以上のごく簡単な例を通して、アフィン演算の雰囲気が少しでも伝わっていたら幸いである。

2.2 アフィン形式

前節で示した例から容易に推測できるように、アフィン演算における変数の一般形は次のようになる：

$$x = x_0 + x_1 \epsilon_1 + x_2 \epsilon_2 + \cdots + x_n \epsilon_n, \quad (-1 \leq \epsilon_i \leq 1, \quad i = 1, \dots, n) \quad (19)$$

式(19)のような変数 x の表し方を x のアフィン形式と言う。計算機には $x_k (k=0, \dots, n)$ の値を記憶させる。

2.3 変数の初期化

アフィン演算では区間 $[a, b]$ を次のように初期化する。

$$\frac{a+b}{2} + \frac{b-a}{2} \epsilon, \quad (-1 \leq \epsilon \leq 1) \quad (20)$$

変数がベクトルや行列である場合は各成分ごとに同様の初期化を行う。ただし、一般に各成分は独立に動くので、各成分ごとに別々の ϵ を用いる必要がある。

2.4 点 (x, y) が取り得る領域 (joint range)

本稿では結果だけ紹介するが、2つのアフィン形式

$$x = x_0 + x_1 \epsilon_1 + \cdots + x_n \epsilon_n \quad (21)$$

$$y = y_0 + y_1 \epsilon_1 + \cdots + y_n \epsilon_n \quad (22)$$

に対して、一般に点 (x, y) の表す領域 (joint range と呼ぶ) は、「点 (x_0, y_0) を対称の中心とする点対称な凸 $2k$ 角形 ($k \leq n$ 、ただし $k=1$ のときは線分)」となる。

なお、このことをくわしく調べる場合、先に2.1節で示した図2などが参考になると思われる。

2.5 通常の区間表示への還元

アフィン形式 x はいつでも通常の区間表示に還元できる(それが得策であるかどうかは別として)。式(19)のアフィン形式を通常の区間表示に還元するには、区間 $[\underline{x}, \bar{x}]$ の形に直すとして $\underline{x}, \bar{x}$ を次のように表せば良い。

$$\underline{x} = x_0 - \Delta_x, \quad \bar{x} = x_0 + \Delta_x, \quad \Delta_x = \sum_{i=1}^n (|x_i|) \quad (23)$$

ここで、 Δ_x は $\text{rad}(x)$ (区間 $[\underline{x}, \bar{x}]$ の半径) に等しい。

2.6 線形演算

アフィン形式 x, y と実数 α を用いて、アフィン演算における線形演算は次のように定義される。

$$x \pm y = (x_0 \pm y_0) + \sum_{i=1}^n (x_i \pm y_i) \epsilon_i \quad (24)$$

$$x \pm \alpha = (x_0 \pm \alpha) + \sum_{i=1}^n x_i \epsilon_i \quad (25)$$

$$\alpha x = (\alpha x_0) + \sum_{i=1}^n (\alpha x_i) \epsilon_i \quad (26)$$

2.7 非線形演算

アフィン演算で難しいのは非線形演算の扱いである。先の2.1節の例で言うと、式(11)のような評価を一体どうやって与えたら良いのかということである。このことを非線形単項演算 ($\sin x, \sqrt{x}$ など) と非線形二項演算 ($x \times y, x^y$ など) に分けて説明する。

2.7.1 非線形単項演算

f を非線形単項演算とすると、アフィン形式 x

$$x = x_0 + x_1 \epsilon_1 + \cdots + x_n \epsilon_n \quad (27)$$

の f による像 $z = f(x)$ は一般に曲線を描くから像そのものをアフィン形式で表すことはできない。

そこで、 x の変域 X において $f(x)$ を

$$ax+b \quad (28)$$

で近似し、そのときの最大絶対誤差 δ を

$$\delta = \max_{x \in X} |f(x) - (ax+b)| \quad (29)$$

で求め、ここで新たなダミー変数 ε_{n+1} を追加して得られる $z=f(x)$ の包含

$$(ax+b) + \delta\varepsilon_{n+1} = (ax_0+b) + \sum_{i=1}^n ax_i\varepsilon_i + \delta\varepsilon_{n+1} \quad (30)$$

を演算結果とする(図3参照)。

図3 非線形単項演算の定義

これは式(27)と比べてダミー変数が1つ増加したアフィン形式である。なお、式(29)の値を厳密に計算することが難しい場合は、かわりにその上界を考える。

式(28)の線形近似における a, b と式(29)の δ の値の取り方によって、最終的な区間評価の性能は大きく異なってくる。その中で、最大絶対誤差 $\delta (\geq 0)$ の値を小さくする評価ほどすぐれた評価であると考えられる。

また、この尺度による最適性に関して、 δ の値が

$$\delta^* = \min_{a, b \in \mathbf{R}} \delta \quad (31)$$

のときの結果が、 $(f(x))$ の近似を式(28)では表せない与え方をした場合も含めてアフィン形式で与え得る最適な定義であることが文献[3]に示されている。

では、その最適な定義を具体的にどうやって得るかということについては、非線形単項演算の種類ごとに個別に定義するよりない。文献[2]には、 $\sqrt{x}, e^x, \log x$ など凸性の変わらない初等関数の定義が示されているが、 $f(x) = \sqrt{x}$ の場合だけ紹介すると、 $x \in [\underline{x}, \bar{x}]$ として

$$\sqrt{x} = \left(\frac{1}{\sqrt{\underline{x}} + \sqrt{\bar{x}}} x + \frac{\underline{x} + \bar{x} + 6\sqrt{\underline{x}\bar{x}}}{8(\sqrt{\underline{x}} + \sqrt{\bar{x}})} \right)$$

$$+ \frac{\underline{x} + \bar{x} - 2\sqrt{\underline{x}\bar{x}}}{8(\sqrt{\underline{x}} + \sqrt{\bar{x}})} \varepsilon_{n+1} \quad (32)$$

のように定義する。この式(32)に

$$[\underline{x}, \bar{x}] = [0.01, 0.49] \quad (33)$$

$$x = 0.25 + 0.24\varepsilon_1 \quad (34)$$

を代入すると、式(12)と同一の結果を得る。

2.7.2 非線形二項演算

f を非線形二項演算とすると、アフィン形式 x, y

$$x = x_0 + x_1\varepsilon_1 + \cdots + x_n\varepsilon_n \quad (35)$$

$$y = y_0 + y_1\varepsilon_1 + \cdots + y_n\varepsilon_n \quad (36)$$

の f による像 $z=f(x, y)$ は一般に曲面を描くから像そのものをアフィン形式で表すことはできない。

そこで、前節と同様に x, y の変域をそれぞれ X, Y として、長方形領域 $X \times Y$ において $f(x, y)$ を

$$ax + by + c \quad (37)$$

で近似し、そのときの最大絶対誤差 δ を

$$\delta = \max_{\substack{x \in X \\ y \in Y}} |f(x, y) - (ax + by + c)| \quad (38)$$

で求め、ここで新たなダミー変数 ε_{n+1} を追加して得られる $z=f(x, y)$ の包含

$$(ax + by + c) + \delta\varepsilon_{n+1} = (ax_0 + by_0 + c) + \sum_{i=1}^n (ax_i + by_i) \varepsilon_i + \delta\varepsilon_{n+1} \quad (39)$$

を演算結果としても良いが、実はこの場合、 δ の値を

$$\delta^* = \min_{a, b, c \in \mathbf{R}} \delta \quad (40)$$

としても、一般には最適な結果を与えない。

これは、2.3節で述べた joint range が一般には長方形領域 $X \times Y$ の一部分しか取らないことと関連している。つまり、joint range D において式(37)の線形近似を求め、そのときの最大絶対誤差 δ を式(38)でなく

$$\delta = \max_{(x, y) \in D} |f(x, y) - (ax + by + c)| \quad (41)$$

のように求めてやれば、 δ の値が式(40)のときに最適な評価を与える。

しかし、非線形二項演算の場合、最適な定義どころか何らかの定義を与えることすら難しく、文献[2]ではわざわざ乗算を定義してあるのみである。それによ

ると、

$$x = x_0 + x_1 \epsilon_1 + \cdots + x_n \epsilon_n \quad (42)$$

$$y = y_0 + y_1 \epsilon_1 + \cdots + y_n \epsilon_n \quad (43)$$

に対して、乗算 $x \times y$ を以下のように定義している：

$$x \times y = y_0 x + x_0 y - x_0 y_0 + \Delta_x \Delta_y \epsilon_{n+1} \quad (44)$$

ただし、 Δ_x , Δ_y はそれぞれ 2.5 節の式 (23) のように定まるアフィン形式 x , y の半径である。

これに対し、文献 [3] [4] では文献 [2] の定義と比べて常に同等以上の結果を与える「改良乗算」が、また文献 [5] ではアフィン形式で与え得る最適な乗算 (理論限界、「最良乗算」と呼ばれる) がそれぞれ提案された。

また、文献 [5] において、乗算 $x \times y$ の最適な傾きは $a = y_0$, $b = x_0$ で与えられることが証明され (c, δ の求め方は煩雑になるので本稿では割愛する)、追加する誤差項の係数 δ (最大絶対誤差) の大小に関して改良乗算は文献 [2] の方法の平均 7 割強、最良乗算は文献 [2] の方法の同 5 割程度であることが示された。

また、文献 [2] では除算を $x \times (1/y)$ のように逆数を取る単項演算と乗算との組み合わせとして実現されていたのに対して、文献 [3] では除算を直接 (1 回の非線形演算として) 定義した。複雑な説明を要するので詳細は割愛するが、2.7.2 節の式 (38) の δ の値を最小にするような定義などが提案されている。

2.8 何に應用するか

さて、確かにシャープな区間評価ができるが計算コストも高そうなこのアフィン演算は、一体どのようなことに役立つのだろうか。有望な応用例として、本稿では非線形方程式の全解探索問題への適用を挙げておく。

この問題は、与えられた領域内において非線形方程式のすべての解を求めることを目的としている。基本的なアルゴリズムは「存在テスト (領域内に解が 1 つだけ存在するための十分条件が成り立つか調べる)」「非存在テスト (領域内に解が 1 つも存在しないための十分条件が成り立つか調べる)」を行い、2 つのテストともに失敗したら領域を分割して再帰的にこれらのテストを行う、というものである。そして、解の非存在領域を効率的に (分割をたくさん行う前に) 見つけることが、問題を高速に解くために重要であることがわかっている。

文献 [6] では、全解探索アルゴリズムの非存在テストにアフィン演算を利用した方法を提案し、数値例によってその有効性を確かめている。また、全解探索ア

ルゴリズムには、文献 [7] で提案された、停止性を考慮したアルゴリズムを採用している。

多岐にわたる実験のごく一部を紹介すると、以下のようなエサキダイオードからなる非線形抵抗回路の回路方程式の全解探索問題

$$\begin{cases} g(x_1) + x_1 + x_2 + \cdots + x_n = 1 \\ g(x_2) + x_1 + x_2 + \cdots + x_n = 2 \\ \vdots \\ g(x_n) + x_1 + x_2 + \cdots + x_n = n \end{cases} \quad (45)$$

に対し、ダイオードの特性を

$$g(x) = 2.5x^3 - 10.5x^2 + 11.8x \quad (46)$$

解の探索範囲を $[-3, 3] \times [-3, 3] \times \cdots \times [-3, 3]$ とした結果、いくつかある提案法のうち最も有力と思われる方法は、 $n=2, 5$ のとき (解の非存在テストにアフィン演算を用いない) 従来の方法のそれぞれ 5%, 1% 程度の分割数、20%, 2% 程度の実行時間で探索範囲内のすべての解を求めることに成功している。

2.9 アフィン演算における丸め誤差の扱い

ここまでの議論では変数が実数であることを想定していたが、これらの計算を計算機に委ねる (べくプログラムを書く) ためにはアフィン演算において計算機の丸め誤差をどう扱うかを考える必要がある。この問題は非常に重要であるが、同時に非常に難しい問題でもある。

現在多くのパソコンやワークステーションでは IEEE 標準 754 という規格に基づいた 2 進の浮動小数点数システムを採用しており、これらの計算機を援用して数値計算を行う場合は、計算の各過程における丸めの向きを制御することができる。

そのため、計算機上で実数を表現するときにはこれを挟む 2 つの浮動小数点数で規定される区間を用い、また区間同士の演算においては下端を下向き丸めで、上端を上向き丸めでそれぞれ計算することが、区間演算を計算機の浮動小数点数システム上で展開することを可能にしていた。

アフィン演算の場合はどうかというと、基本的にはアフィン形式の係数 x_i 各々に対してこれらをいったん区間 $[x_i, \overline{x_i}]$ のような形で求めておき、

$$x_i = \frac{x_i + \overline{x_i}}{2} \quad (47)$$

を上向き丸めで計算することで最終的な係数 x_i を与えるとともに、

NII-Electronic Library Service