区間演算システムの構築

2003MM089 大澤 智弘
指導教員: 杉浦 洋

1 はじめに

現代のコンピュータにおいて，高速な数値計算ができるのは，実数を浮動小数点数で近似して計算を行うからである．浮動小数点数とは，規格化された有限桁の小数である．その計算結果は，近似であるため厳密に正しいことが保証されていない．そこで，精度保証付き数値計算．つまり数値計算結果がどのくらい正しいのか検算することが重要となってくる．本研究は，精度保証付き数値計算システムの構築を目的として行う．

2 IEEE754

IEE754とは，コンピュータやワークステーションなどで標準的に用いられている浮動小数点数システムの規格である．IEEE754とは，コンピュータやワークステーションなどで標準的に用いられている浮動小数点数システムの規格である．IEEE754は，浮動小数点集合の定義と，その上の四則演算と開平の規格からなる．大きな特徴は，丸めモードが選択できることで，これにより四則演算と開平の精度保証が可能となる．しかし，多様な精度保証付き数値計算を効果的に行うには，初等関数の区間関数を開発する必要がある[1,2]．

2.1 浮動小数点数の規格

２進規格化浮動小数点数とは，0と

$$a = \pm\left(\frac{1}{2} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + \cdots\cdots + \frac{d_N}{2^N}\right) \times 2^e, \quad d_i \in \{0,1\},$$

と書ける数をいう．小数部の桁数Nはシステムの定数で，倍精度では$N = 53$である．２進規格化浮動小数点数全体の集合を$\mathbf{F}$とする．

2.2 丸めモード

計算機上では，一般の実数は，$\mathbf{F}$の数に丸められ，メモリに格納される．IEEE754では次の４つの丸めモードが指定できる．cを実数($c \in \mathbf{R}$) とし以下にまとめる．

1. 上向き丸め(round upward) c以上の最小の浮動小数点数に丸める．これを$\triangle : \mathbf{R} \to \mathbf{F}$と表す．
2. 下向き丸め(round downward) c以下の最大の浮動小数点数に丸める．これを$\nabla : \mathbf{R} \to \mathbf{F}$と表す．
3. 最近点への丸め(round to nearest) cに最も近い浮動小数点数に丸める．これを$\square : \mathbf{R} \to \mathbf{F}$と表す．もし，このような点が２点ある場合には，仮数部の最後のビットが0である浮動小数点数に丸める．
4. 切捨て(round toward 0) 絶対値が$|c|$以下の浮動小数点数の中でcに最も近いものに丸める．

2.3 機械四則演算の規格

機械四則演算は，真の四則演算の結果を丸めたものと完全に一致するように定められている．すなわち，指定された丸め$\bigcirc \in \{\triangle, \nabla, \square\}$における，真の四則演算$\cdot \in \{+, -, \times, /\}$に対応する機械四則演算$\odot$は

$$x \odot y = \bigcirc(x \cdot y) \quad (x, y \in F)$$

である．

3 区間解析

区間解析において，区間とは，閉区間

$$[\underline{x}, \overline{x}] = \{x \in \mathbf{R} \mid \underline{x} \le x \le \overline{x}\}$$

である．ただし，$\underline{x} \le \overline{x} \in \mathbf{R}$．区間を$[x] = [\underline{x}, \overline{x}]$と表すこともある．区間全体の集合を$\mathbf{IR}$と書く．

3.1 区間四則演算

二つの区間$[x], [y]$が与えられたとき，その二つの区間の四則演算を次のように定義する．

$$[x] \circ [y] = \{x \circ y | x \in [x], y \in [y]\}$$

ただし，$\circ \in \{+, -, \times, /\}$である．これを区間演算といい，次の式が成立する．

1. $[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
2. $[x] - [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$
3. $[x] \times [y] = [\min\{\underline{x}\underline{y}, \overline{x}\overline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}\}, \max\{\underline{x}\underline{y}, \overline{x}\overline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}\}]$
4. $[x]/[y] = [x] \times [\frac{1}{\overline{y}}, \frac{1}{\underline{y}}]$, $(0 \notin [y])$

3.2 区間関数

関数 f:$D \subset \mathbf{R} \to \mathbf{R}$ を領域Dの任意の閉区間の上で連続な関数とする．関数fを区間関数

$$f([x]) = \{f(x) | x \in [x]\}$$

により$\mathbf{IR}$上の関数に拡張できる．

3.2.1 初等関数の区間関数

初等関数は性質がよく分かっているので以下のように簡単に区間関数を構成できる．

$$f([x]) = [f(\underline{x}), f(\overline{x})], \quad (f \in \arctan, \arcsin, \sinh, \tanh, \exp, \log)$$

$$f([x]) = [f(\overline{x}), f(\underline{x})], (f \in \operatorname{arccot}, \operatorname{arcoth})$$

3.2.2 合成関数の区間拡張

初等関数と四則演算による合成関数をfとする．fを構成する初等関数と四則演算を全てその区間関数で置き換えたものをfの区間拡張といい$f_{[]}$で表す．このとき

$$f([x]) \subseteq f_{[]}([x])$$

が成立する．また，上式で等式が成立するとき，厳密な包み込みができたという．

3.3 区間包囲

関数の値域$f([a,b])$を$f([a,b]) \subset [c,d]$と区間$[c,d]$で評価するとき，区間$[c,d]$のことを$f([a,b])$の区間包囲という．式の表現に$[x]$が少なく現れれば現れるほど，区間包囲は厳密な包み込みに近くなることが多い．

4　機械区間演算

区間演算においては，区間の両端の数は実数であるとし，厳密な実数演算に基づいて区間演算が定義された．しかし，浮動小数点数システム上で数値計算するときは，丸めが起こるため，厳密な区間演算は実行できない．そこで，区間演算を与えられた浮動小数点数システム$\mathbf{F}$上に展開する方法について述べる．両端を浮動小数点数全体とする$\mathbf{IF}$を

$$\mathbf{IF} = \{[\underline{x}, \overline{x}] \in \mathbf{IR} | \underline{x}, \overline{x} \in \mathbf{F}\}$$

とする．$\mathbf{IF}$の要素を機械区間という．

機械区間$[x], [y] \in \mathbf{IF}$と四則演算・$\in \{+, -, \times, /\}$に対し，区間四則演算の結果を$[z] = [x] \cdot [y] \in \mathbf{IR}$とする．3.1節の定義に基づき，下向き丸めモードで下限$\underline{z}$を，上向き丸めモードで$\overline{z}$を計算すれば，

$$[z'] = [x] \odot [y] \triangleq [\nabla \underline{z}, \triangle \overline{z}] \in \mathbf{IF}, [z'] \supset [z]$$

を計算することができる．これを機械区間四則演算という．

機械区間はC++のクラスとして実現する．また，区間演算，区間初等関数はそのクラスの関数として実装し，元の関数にオーバーレイする．このことにより，区間計算はC++プログラムで

$$y = x + \sin(x); \quad (x, y\text{は区間})$$

のように書ける．

5　区間指数関数

平均値の定理により次の定理を得る．

定理 1　区間$[a, b]$におけるC^1級関数を$f(x)$とする．区間のn等分点を

$$x_i = a + ih \ \ (0 \le i \le n) \ , \ \ h = \frac{b-a}{n},$$

$$\begin{cases} \max\limits_{0 \le i \le n} |f(x_i)| \le \epsilon_0 \\ \max\limits_{a \le x \le b} |f'(x)| \le \epsilon_1 \end{cases}$$

とすると

$$\max_{a \le x \le b} |f(x)| \le \epsilon_0 + \frac{h}{2}\epsilon_1$$

5.1　近似式の導出

まず，$f(x) = e^x, x \in \mathbf{F} \cap (0, \infty)$の包囲区間$F(x) \in \mathbf{IF}$ を精度よく求める方法を述べる．$d = 2^{-5}$とすると，$4.94 \times 10^{-324} < x < 1.80 \times 10^{308}$ゆえ，

$$x = 8n + \frac{k}{16} + r, -94 \le n \le 88, 0 \le k < 128, |r| \le d$$

と書ける．ゆえに，$e^{8n} \in [a_n](-94 \le n \le 88), e^{\frac{k}{16}} \in [b_k](0 \le k < 128)$なる区間をあらかじめ計算しデータとして保持すれば，

$$f(x) = e^{8n} e^{\frac{k}{16}} e^r \in F(x) \triangleq [a_n] \otimes [b_k] \otimes F(r)$$

となり，問題は$e^r(|r| \le d)$の包囲区間$F(r)$の計算に局限される．これを区間縮小法という．

ここで$f(-r) = \frac{1}{f(r)}$なる性質に着目して

$$g(r) = \frac{p(r)}{p(-r)}$$

とし，$p(r)$をn次多項式とする．最大絶対誤差が

$$\max_{|r| \le d} |f(r) - g(r)| \le 5 \times 10^{-18}$$

となることを目標に試行錯誤した結果，$p(r)$の次数は3で十分であることが分かった．これにより

$$g(r) = \frac{p(r)}{p(-r)} = \frac{1 + ar + br^2 + cr^3}{1 - ar + br^2 - cr^3} \cong e^r \qquad (1)$$

とする．係数a, b, cは残差

$$e(r) = p(-r)e^r - p(r)$$

の絶対値がなるべく小さくなるように設計する．

Mathematicaによる多倍長区間演算で

$$|e'(r)| < \epsilon_1 = 10^{-12} \quad (|r| \le d)$$
$$\max_{0 \le i \le n} |e(r_i)| \le \epsilon_0 = 4.7 \times 10^{-18}$$

を得た．ただし，区間$[-d, d]$の$n = 4096$等分点を$r_i = -d + ih(0 \le i \le n)$ とする．定理1より，

$$|e(r)| \le \delta_r = \epsilon_0 + \frac{h}{2}\epsilon_1 \ \ (|r| \le d)$$

となる．結果，今回得た係数では

$$\left| \frac{f(r) - g(r)}{g(r)} \right| \le \epsilon = 5 \times 10^{-18} \quad (|r| \le \frac{1}{32}) \qquad (2)$$

となった．

5.2　区間関数

区間$e^{8n} \in [a_n](-94 \le n \le 88), e^{\frac{k}{16}} \in [b_k](0 \le k < 128)$はMathematicaの多倍長区間演算で精度よく求め，それを拡張倍精度に丸めてデータとした．指数関数は単調増加だから，前節で設計した関数$F(x)$により，区間指数関数は

$$F([x]) \triangleq \left[\underline{F(\underline{x})}, \overline{F(\overline{x})}\right]$$

で計算でき，$e^{[x]} \in F([x])$となる．

6　おわりに

今回の研究では，区間対数関数，区間指数関数の実装を完了することができた．今後の課題は，残りすべての初等関数の区間関数を実現することである．

参考文献

[1] 大石進一：数値計算, 裳華房 (1999).

[2] 大石進一：精度保証付き数値計算, コロナ社 (2000).