精度保証付き数値積分

2006MI144 斎藤裕樹
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．ただし，浮動小数点数とは規格化された有限桁の小数であるので，その数値計算の結果は近似であり，数学的な意味では厳密に正しいと保証されていない．そこで，この数値計算結果がどのくらい正しいのかを検算する．これが精度保証付き数値計算である．精度保証付き数値計算の基本的なアイデアは，浮動小数点数を両端とする区間の中に連続数学の問題の解を包み込もうというものである．

本研究では，Mathematica による精度保証付き数値積分を計算するプログラムを作成した．

2　浮動小数点数

浮動小数点数システムの規格として，IEEE754 が多くのコンピュータで標準的に用いられている．本研究では IEEE754 に基づく 2 進浮動小数点数システムを利用する．2 進規格化浮動小数点数とは，0 と

$$a = \pm(\sum_{k=0}^{N} 2^{-k} d_k) \times 2^e, d_i \in \{0,1\}$$

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．このとき，$-1022 \leq e \leq 1023$ である．2 進規格化浮動小数点数全体に 0 を合わせて機械実数と呼ぶ．機械実数全体の集合を $\mathbb{F}$ とする．

3　区間解析

3.1　区間

区間解析において，区間とは，閉区間

$$[\underline{x}, \overline{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \overline{x}\}$$

である．区間を $[x] = [\underline{x}, \overline{x}]$ と表すこともある．区間全体の集合を $\mathbb{IR}$ と書く．$\underline{x} = \overline{x}$ となる区間 $[x]$ は点区間という．

3.2　区間関数

関数 $f : D \subset \mathbb{R} \to \mathbb{R}$ を領域 D で連続な関数とする．関数 f を区間関数

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

により $\mathbb{IR}$ 上の関数に拡張できる．

標準関数と四則演算による合成関数を f とする．f を構成する標準関数と四則演算をすべ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．

4　定積分の数値計算

4.1　複合台形公式と誤差公式

区間 $[a,b]$ 上に等間隔に分点 $t_i = a + ih (0 \leq i \leq n), h = \frac{b-a}{n}$ をとり，複合台形公式を適用すると

$$I = \frac{h}{2}\{f_0 + 2f_1 + \cdots + 2f_{n-1} + f_n\} + E,$$

$$E = -\frac{h^3}{12}\{f^{(2)}(\xi_1) + \cdots + f^{(2)}(\xi_n)\}$$

となり，誤差 E は

$$| E | \leq \frac{(b-a)^3}{12n^2} \| f^{(2)} \|$$

と評価できる．ここで，$\|f\| = \max_{x \in [a,b]} |f(x)|$ は一様ノルムである．

4.2　計算手順

精度保証付き数値積分の手順を以下に表す．

1. $T_n = \frac{h}{2}\{f_0 + 2f_1 + \cdots + 2f_{n-1} + f_n\}$ を区間演算する．$T_n \in [\underline{T}_n, \overline{T}_n]$
2. 2 階導関数のノルムの上界 M を計算する．
   $\| f^{(2)} \| = \max\limits_{a \leq x \leq b} | f^{(2)}(x) | \leq M$
3. 最大誤差 $\overline{E} = \frac{(b-a)^3}{12n^2} M$ を計算する．
4. 定積分値 I の包囲区間 $[I] = [\underline{T}_n - \overline{E}, \overline{T}_n + \overline{E}]$ を計算する．

4.3　関数の最大値の計算法

M を求めるために，与えられた区間 $[x]$ における関数 $f(x)$ の最大値を求めるアルゴリズムが必要である．f を $[x]$ で 2 回連続微分可能な関数とするとき，$f^* = \max\limits_{x \in [x]} f(x)$ とそれを達成する x^* をすべて求める手続きを以下に述べる．

1. リスト L を用意して初期値を L= $([x])$ とする．
2. 細分：リストから区間を取り出し，分割して部分区間をリストに戻す．L= $([x])$ →L= $([y_1], [y_2], \cdots, [y_n])$
3. 廃棄：最大値を含まないことが確定した区間をリストから廃棄する．

リスト内の区間の直径 ϵ が 10^{-6} 以下になるまで 2,3 を繰り返す．

4.4　廃棄基準

4.4.1　Midpoint Test

初期値を $\tilde{f} = +\infty$ とし，上端が最大の部分区間 $[y_z]$ を L から選ぶ．つまり，$\overline{f([y_z])} = \max\limits_{[y_k] \in \mathrm{L}} \overline{f([y_k])}$ とする．さらに，$c = m([y_z]), \tilde{f} = \min\{f(c), \tilde{f}\}$ とし，$\overline{f([y_k])} < \tilde{f}$ となる部分区間 $[y_k]$ をリストから廃棄できる．

4.4.2 Monotonicity Test

f が部分区間 $[y_k]$ で狭義単調ならば，f は部分区間 $[y_k]$ で最大値をとりえない．つまり，$0 \notin f'([y_k])$ ならば，その部分区間 $[y_k]$ をリストから棄却できる．ただし例外として，部分区間 $[y_1]$ で単調減少の場合は左端点区間 $[\underline{x}, \underline{x}]$ を残し，また，部分区間 $[y_n]$ で単調増加の場合は右端点区間 $[\overline{x}, \overline{x}]$ を残す．

4.4.3 Convexity Test

f が部分区間 $[y_k]$ で下に凸ならば，f は部分区間 $[y_k]$ で最大値をとりえない．つまり，$\underline{f''([y_k])} > 0$ ならば，その部分区間 $[y_k]$ をリストから棄却できる．ただし例外として，$\underline{f''([y_1])} > 0$ である場合でも左端点区間 $[\underline{x}, \underline{x}]$ を残し，また，$\underline{f''([y_n])} > 0$ である場合でも右端点区間 $[\overline{x}, \overline{x}]$ を残す．

5 実験結果

台形公式の分割数 n を 10 とし，廃棄基準の分割数 $n2$ は 2,4,8,16 とし，区間 $[0, 1]$ において，以下の 2 式について実験を行った。

$$f_s(x) = (\sin \pi x)(\sin ax), f_e(x) = e^{ax} \quad (a = 1, 2, 4, 8, 16)$$

$[I]$：定積分値の包囲区間
I：真の積分値
$\overline{E}$：台形公式による最大誤差

5.1 $f_s(x) = (\sin \pi x)(\sin ax)$

実験を始める際に $f_s(x)$ のグラフを見て，極点の数で分割することが最も効率の良い方法ではないのかと考え，対応する極点の数を $n2$ に加えて実験を行った．

表 1 $f_s(x)$ の呼び出し回数（単位：回）

a ＼ n2	2	3	4	7	8	16
1	451	—	336	—	376	464
2	463	—	331	—	370	458
4	495	416	370	—	437	544
8	529	—	403	—	441	547
16	582	—	453	472	497	556

表 2 $f_s(x)$ の近似積分値

a	$[I]$	I	$\overline{E}$
1	{ 0.288,0.303}	0.299	0.0072
2	{ 0.472,0.496}	0.487	0.012
4	{ 0.368,0.412}	-0.388	0.022
8	{ -0.13,0.0007}	-0.058	0.061
16	{ -0.22,0.23}	0.0037	0.23

以上より，$a = 1, 2, 4, 8, 16$ のすべての場合において，4 分割したときが効率が良いという結果が出た．また，$n2$ によらず，近似積分値はほぼ同じであった．さらに，近似積分値は a が大きいほど，精度が悪くなることがわかった．

5.2 $f_e(x) = e^{ax}$

実験結果は以下の通りである．

表 3 $f_e(x)$ の呼び出し回数（単位：回）

a ＼ n2	2	4	8	16
1,2,4,8,16	37	45	61	93

表 4 $f_e(x)$ の近似積分値

a	$[I]$	I	$\overline{E}$
1	{ 1.717,1.722}	1.719	0.0023
2	{ 3.18,3.23}	3.20	0.025
4	{ 12.8,14.4}	13.4	0.73
8	$\{2.3 \times 10^2, 5.6 \times 10^2\}$	3.8×10^2	1.6×10^2
16	$\{ -1.2 \times 10^6, 2.6 \times 10^6\}$	5.6×10^5	1.9×10^6

$n2$ によらず，近似積分値は同じであった．さらに，a がどの値でも，$f_e(x)$ の呼び出し回数が同じになった．これは，a によらず，$f_e''(x)$ がすべて右上がりのグラフになり，Monotonicity Test または，Convexity Test のみで最大値をとる部分区間（今回は右端点区間）が決まるので，$f_e(x)$ の呼び出し回数は a に依存しない．ゆえに，廃棄基準を Monotonicity Test のみでの実験も行った．結果は，呼び出し回数は予想通り減少し，近似積分値は同じ結果になった．

$f_s(x)$，$f_e(x)$ ともに，目的としていた $I \in [I]$ を達成した．また，複合台形公式の誤差に比べ，丸め誤差はかなり小さいものであるので，$\overline{E}$ は $[I]$ の半径にほぼ等しいという結果になった．

6 おわりに

本研究では，Mathematica による精度保証付き数値積分を計算するプログラムを実装した．この複合台形公式は，図形が複雑になると精度が非常に悪くなるというものであったが，分割数を様々な値で実験でき，台形公式よりも面白いものであった．

今後の課題は，終了条件 ϵ の値をさらに小さくすることや，台形公式に用いる分割数 n を大きくすることで，さらに精度を高めること，また，様々な関数の波形によって 3 つの廃棄基準の順序を考え，計算コストの効率化に関する研究が必要である．

参考文献

[1] 大石進一：精度保証付き数値計算，コロナ社 (2000).

[2] 大澤智弘：区間演算システムの構築，南山大学数理情報学部数理科学科卒業論文 (2006).

[3] 杉浦洋：数値計算の基礎と応用－数値解析学への入門－，サイエンス社 (1997).

[4] R. Hmmer, M. Hocks, U. Kulisch and D. Ratz: *Numerical Toolbox for Verfied Computing I -Basic Numerical Problems-*. Springer-Verlag, New York, 1993.