数理解析研究所講究録
928巻 1995年 38-41

区間演算による関数の値域の効率的評価法

Effective Interval Evaluation of the Range of Functions

九州大学 工学部 情報工学科　柏木 雅英 (Masahide Kashiwagi)

1 はじめに

区間演算は、数値を [下端, 上端] という閉区間で表現し演算を行っていく方法で、本来は、数値計算の際に発生する丸め誤差を把握するために考案されたものである。しかし、一方で関数に区間を入力すると入力区間の像の包み込みが区間として得られ、関数の値域を容易に評価出来るという利点もある。

一方、区間演算の欠点として、評価が'悲観的'に過ぎ、出力される区間幅が非現実的に大きくなってしまい役に立たないことも多いということが指摘されている。これの解決法として、平均値の定理を用いる方法 (mean value form) が知られている。これは、入力区間 I の関数 f による像 $f(I)$ を評価するために、f の演算過程を単純に区間演算に置き換えるのでなく、f の導関数 f' の計算過程を区間演算に置き換えた区間写像 F' を用いて、

$$f(c) + F'(I)(I - c) \tag{1}$$

とする方法である (c は I の中心)。この方法を用いるためには、導関数 f' の計算過程が必要であるが、これは数式処理などを用いるのではなく、自動微分の計算過程を区間演算に置き換えることによって $F'(I)$ を得ると効率がよいことが知られている (例えば [1])。

ところで、この $F'(I)$ を Bottom Up 型の自動微分によって計算する場合、途中に現れる変数 (これは入力区間 I の関数である) における入力区間 I の像の区間評価が利用される。これは、その変数の計算過程を単純に区間演算に置き換えたものによって計算されるが、これに対して mean value form を適用すればより良い評価が得られることが分かる。本稿では、Bottom Up 型の自動微分を行うのとほぼ同様の方法で、関数の計算過程に現れる全ての変数に対して mean value form を適用できるような新しい区間演算のアルゴリズムについて報告する。この方法は、単純区間演算、区間演算に置き換えた Bottom Up 型自動微分の双方に比べて、少なくとも評価が悪くなることはなく、大抵は評価が良くなる。

2 対象とする関数

関数 $f : \boldsymbol{R}^n \to \boldsymbol{R}^m$ の内で、

$$z_i = \begin{cases} g_i(z_{a_i}, z_{b_i}) \\ \text{or} \\ h_i(z_{a_i}) \end{cases} \qquad (n+1 \leq i \leq l) \tag{2}$$

のように、二項演算 g (加減乗除、累乗など) または単項演算 h (sin , log など) の組み合わせで記述されている関数を対象とする。ここで、変数 $z_1 \cdots z_n$ は入力変数 $x_1 \cdots x_n$ に対応し、$1 \leq a_i, b_i < i$ とする。$z_{n+1} \cdots v_l$ のうちの m 個は、関数 f の出力に対応する。また、ここで現れる演算は全て微分可能で、かつ演算自身及び演算の微分に対する区間演算が可能であるものとする。

3 アルゴリズム

アルゴリズムは、次の通りである。

単純に自動微分を区間演算化した場合、各変数が保持する量は、その変数を入力 $x \in \boldsymbol{R}^n$ の関数として $z(x)$ と書くと、I を入力区間ベクトルとして $V \supset \{z(x)|x \in I\}$ と、$D \supset \{z'(x)|x \in I\}$ の二つである。本稿の方法では、その二つに加えて、I の中心 c の像 $v \ni z(c)$ を保持して演算を行う。

まず、入力変数 $z_1 \cdots z_n$ を

$$V_i = I_{(i)} \tag{3}$$
$$v_i = c_{(i)} = \mathrm{mid}(I_{(i)}) \tag{4}$$
$$D_i = (e_1 \cdots e_n) \quad (e_j = \delta_{ij}) \tag{5}$$

で初期化する。ここで、区間ベクトル $I-c$ の値を以後の演算で使用するために記憶しておく。二項演算は、$z_i = g_i(z_{a_i}, z_{b_i})$ に対して、

$$v_i = g_i(v_{a_i}, v_{b_i}) \tag{6}$$
$$D_i = \frac{\partial g_i}{\partial z_{a_i}}(V_{a_i}, V_{b_i})D_{a_i} + \frac{\partial g_i}{\partial z_{b_i}}(V_{a_i}, V_{b_i})D_{b_i} \tag{7}$$
$$V_i = g_i(V_{a_i}, V_{b_i}) \cap \{v_i + D_i(I-c)\} \tag{8}$$

で行う。単項演算は、$z_i = g_i(z_{a_i})$ に対して、

$$v_i = g_i(v_{a_i}) \tag{9}$$
$$D_i = g_i'(V_{a_i})D_{a_i} \tag{10}$$
$$V_i = g_i(V_{a_i}) \cap \{v_i + D_i(I-c)\} \tag{11}$$

で行う。

例えば、減算 ($z_i = z_{a_i} - z_{b_i}$) の場合は、

$$v_i = v_{a_i} - v_{b_i}$$
$$D_i = D_{a_i} - D_{b_i}$$
$$V_i = (V_{a_i} - V_{b_i}) \cap \{v_i + D_i(I-c)\}$$

となり、乗算 ($z_i = z_{a_i} \times z_{b_i}$) の場合は、

$$v_i = v_{a_i} \times v_{b_i}$$
$$D_i = V_{b_i}D_{a_i} + V_{a_i}D_{b_i}$$
$$V_i = (V_{a_i} \times V_{b_i}) \cap \{v_i + D_i(I-c)\}$$

となり、sin ($z_i = \sin(z_{a_i})$) の場合は、

$$v_i = \sin(v_{a_i})$$
$$D_i = \cos(V_{a_i})D_{a_i}$$
$$V_i = \sin(V_{a_i}) \cap \{v_i + D_i(I-c)\}$$

となる。

4 例 1

例として、関数 $f(x) = (8x - x^2 - 16)(x - 3)$ の値を入力区間 $I = [3, 5]$ で評価してみる。単純区間演算、区間演算に置き換えた Bottom Up 型自動微分、本報告の方法の 3 つの場合について、計算経過を表 1に示す。 自動微分法の $f(I)$ の評価は、mean value form を用いて復元したもの

表 1: 計算経過の比較

計算方法	単純区間演算	自動微分	新方式
表示形式	V	$V\langle D\rangle$	$V\{v\langle D\rangle\}$
x	$[3,5]$	$[3,5]\langle 1\rangle$	$[3,5]\{4\langle 1\rangle\}$
$8x$	$[24,40]$	$[24,40]\langle 8\rangle$	$[24,40]\{32\langle 8\rangle\}$
x^2	$[9,25]$	$[9,25]\langle[6,10]\rangle$	$[9,25]\{16\langle[6,10]\rangle\}$
$8x - x^2$	$[-1,31]$	$[-1,31]\langle[-2,2]\rangle$	$[14,18]\{16\langle[-2,2]\rangle\}$
$8x - x^2 - 16$	$[-17,15]$	$[-17,15]\langle[-2,2]\rangle$	$[-2,2]\{0\langle[-2,2]\rangle\}$
$x - 3$	$[0,2]$	$[0,2]\langle 1\rangle$	$[0,2]\{1\langle 1\rangle\}$
$(8x - x^2 - 16)(x - 3)$	$[-34,30]$	$[-34,30]\langle[-21,19]\rangle$	$[-4,4]\{0\langle[-6,6]\rangle\}$
$f(I)$ の評価	$[-34,30]$	$[-21,21]$	$[-4,4]$
$f'(I)$ の評価	—	$[-21,19]$	$[-6,6]$

である。真の像は、 $f(I) = [-2, 0]$、 $f'(I) = [-5, 1/3]$ である。この関数は評価区間内に 2 つの極値を持つため評価が難しいが、新方式はかなり sharp な評価が行えていることが分かる。

$8x - x^2 - 16$ の時点では、自動微分と新方式で微分の評価 (D) は同じ $[-2, 2]$ であり、 V の評価を自動微分の場合について mean value form を用いて行うと、新方式と同じ $[-2, 2]$ が得られる。ところが、以降の計算においてより悪い評価 $[-17, 15]$ が使用されてしまうため、評価が悪くなる。新方式は、この点が改善されていることが分かる。

5 例 2

次に、文献 [2] で紹介されている Interval Slope による方法との比較を行なう。これは、本手法と全く同じ目的の方法である。その文献による例題: 関数

$$f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1) f_2(x_2) f_3(x_3) f_4(x_4) f_5(x_5) \tag{12}$$

$$f_1(x_1) = 0.01x_1(x_1 + 13)(x_1 - 15) \tag{13}$$
$$f_2(x_2) = 0.01(x_2 + 15)(x_2 + 1)(x_2 - 8) \tag{14}$$
$$f_3(x_3) = 0.01(x_3 + 9)(x_3 - 2)(x_3 - 9) \tag{15}$$
$$f_4(x_4) = 0.01(x_4 + 11)(x_4 + 5)(x_4 - 9) \tag{16}$$
$$f_5(x_5) = 0.01(x_5 + 9)(x_5 - 9)(x_5 - 10) \tag{17}$$

を、

$$x_1 \in [8.7, 8.8]$$
$$x_2 \in [-9.4, -9.3]$$

$$x_3 \in [-4.6, -4.5]$$
$$x_4 \in [3.5, 3.6]$$
$$x_5 \in [-2.9, -2.8]$$

で評価する問題について、本稿の手法と比較した結果を表2で示す。これを見ると、どちらも単

表 2: 出力区間の比較

方法	出力区間	区間幅
単純区間演算	[22283.5, 26731.4]	4447.9
Interval Slope	[24315.0, 24513.7]	198.7
新方式	[24389.0, 24439.7]	50.7

純区間演算に比べると大幅に改善されているが、本稿の手法ではより狭い区間が得られていることが分かる。

6 むすび

評価したい関数 f が1変数の場合は、高次の Taylor 展開と区間演算を組み合わせて非常に sharp な評価が行えることが知られている。しかし、多変数の場合は Taylor 展開を用いるのは実用的でなく、本報告の手法は多変数の場合に容易に適用可能であると言う点で有用であると言える。

謝辞

日頃御指導頂く早稲田大学大石進一、堀内和夫両教授に深謝する。

参考文献

[1] L. B. Rall : "Validation of Numerical Computation", 京都大学数理解析研究所講究録 673, pp.1–16 (1988).

[2] H. C. Fischer : "Automatic Differentiation and Applications", Scientific Computing with Automatic Result Verification (eds. E. Adams and U. Kulisch), Academic Press, pp.105–142 (1993).