区間解析による多峰性関数の極値探索法

藤井康雄　　　　市田浩三
（京大・情報処理教育センター）　（京産大・経営）

1. はじめに

多変数関数の最適化の手法として，現在さまざまな方法が開発されている．関数が単峰性である場合には共役勾配法，シンプレックス法，ダビドンの方法[1]など数多くの方法が発表されている．しかし，関数が多峰性で極値が複数個存在するような場合，常に大域的な最大，最小を求める方法は現在のところ見当らない．決定論的な方法では，複数個の初期推定値から出発して極値を求めるアルゴリズムを繰返し適用して大域的解を推定するのが通常である．また確率論的な方法としては，ランダムサンプリング法，多重積分を利用する方法[2]，クラスタリング法などが有効であると言われている．さらにBremermannは，ランダムベクトル上で4次近似をおこなう方法を提案し，それが100変数以上の場合でも適用できると述べている[3]．これらの方法では，大域的解に到達する可能性があるが，多数回の試行が必要となる．

ここでは、数値を常に幅をもった区間として扱う演算，すなわち「区間演算」[4]に着目し，ある領域（区間）内における大域的な最大，最小を求める方法について述べる．この方法は，評価関数を区間に拡張し，変数領域を順次分割して各部分領域における関数値の上，下限を推定し，最適解を含む可能性の無い領域を捨てていくことにより最大値を求めるアルゴリズムである．この方法は単純であるが正確に最大値に到達する．このアルゴリズムの過程で，解を含む可能性のある残存領域の集りを，それぞれ独立した部分群として抽出し，それぞれの部分領域に区間解析用のNewton法を適用することができる．これは、より多変数の場合に効果がある．

さらに，変数領域の分割と区間解析用のNewton法を組み合せて，最適解をもたない可能性のある領域をできる限り早期（領域が大きい時点）に捨て，かつ残存の個々の領域にNewton法を適用して解を得る方法について述べる．

2. 区間演算

2.1 区間演算の定義[4]

いまある任意の数を，それより小さい有限桁数と，それより大きい有限桁数との組で表し，取扱うすべての数値を幅をもった数値として取扱い，これを「区間」と呼び

$$[a, b] = \{ x \mid a \leq x \leq b \} \qquad とくに\ [a, a] = a \tag{1}$$

とするとき，これらの区間の間の基本演算を次のように定義する．

$$[a, b] + [c, d] = [a+c,\ b+d],$$
$$[a, b] - [c, d] = [a-d,\ b-c],$$

$$\left.\begin{aligned}[a, b]\cdot[c, d]&=[\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)],\\ [a, b]/[c, d]&=[a, b]\cdot[1/d, 1/c]\\ &=[\min(a/d, a/c, b/d, b/c), \max(a/d, a/c, b/d, b/c)],\end{aligned}\right\} \quad (2)$$

ただし $0\notin[c, d]$

(2)の右辺の区間は，計算機による区間演算では，下限は切り捨て，上限は切り上げて，より広い区間に修正される．こうして得られた区間は，無限桁演算による正確な値と，通常の丸めを含んだ計算結果を含んでいる．そのため正確な値と計算機による演算結果との誤差が評価できることになる．

ここで使用した区間演算システムは，FORTRANとアセンブラで作成しており任意の2進桁での切り上げ，切り捨てができるシステムとして構築されている．最近では，小型計算機上にマイクロプログラム方式で作成されたものもある．

この区間演算だけでは，誤差限界は一般には過大評価となるため，区間幅拡大を防止する区間演算アルゴリズムが開発されている．

2.2 区間幅拡大を防ぐアルゴリズム

(2)の区間演算を用いて関数値を評価する場合，前述のように区間演算で求めた評価は，正確な関数値の範囲より拡大する．このため次のような方法を並用して区間幅拡大を防ぐのが一般である．

i) 巣形（nested form）

多項式

$$f(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_0 \quad , \quad a_i\in A_i \ , \ x\in X \quad (3)$$

（ここで A_i, X はそれぞれ a_i, x の区間）

の評価を区間関数で考えるとき，

$$F(X)=(\cdots\cdots((A_n)X+A_{n-1})X+\cdots A_1)X+A_0 \quad (4)$$

なる"巣形"にした方が(3)をそのまま区間関数にした

$$F(X)=A_n X^n+A_{n-1}X^{n-1}+\cdots\cdots+A_0 \quad (5)$$

より $F(X)$ の区間幅は狭くなる[4]．

ii) 中心形（centered form）

多変数関数 $f(x_1, x_2, \cdots, x_n)$ を実有理関数とし，実数 $\mathbf{c}=(c_1, c_2, \cdots, c_n)$ を f の値が定義されるベクトルで

$$f(x_1, x_2, \cdots, x_n)=f(c_1, c_2, \cdots, c_n)+g(x_1-c_1, \cdots, x_n-c_n) \quad (6)$$

と表せるとする．区間 $[a, b]$ の中点は，$m([a, b])=(a+b)/2$ で表されることより，$c_i\ (i=1, \cdots, n)$ として区間 X_i の中点 $m(X_i)$ を用いれば

$$\begin{aligned}F(X_1, X_2, \cdots, X_n)=&f(m(X_1), m(X_2), \cdots\cdots, m(X_n))\\ &+G(X_1-m(X_1), \cdots\cdots, X_n-m(X_n))\end{aligned} \quad (7)$$

となって $F(X_1, X_2, \cdots, X_n)$ をそのまま計算するより狭い区間を得ることができる[4]．

iii) 2乗演算の処理

x^2 すなわち区間演算で $A\cdot A=[c, d]$ のとき, $c<0$ ならば $c=0$ とおき区間関数値の下限を引き上げる. この方法と他の方法を組み合わせて使用すれば効果が大きい.

iv) 区間の細分化

いま X の区間が $[a, b]$ で与えられるとき,

$$[a, b]=[a, a_1]\cup[a_1, a_2]\cup\cdots\cdots\cup[a_k, b] \tag{8}$$

のように X を $k+1$ 等分したものであって, f を実関数とすると

$$\{f(x)\mid x\in[a, b]\}=\bigcup_{i=0}^{k}\{f(x)\mid x\in[a_i, a_{i+1}]\}, \quad (a_0=a, a_{k+1}=b) \tag{9}$$

である. 有理区間関数

$$\bigcup_{i=0}^{k} F([a_i, a_{i+1}])\supset\{f(x)\mid x\in[a, b]\} \tag{10}$$

と表せる. ここで $k\to\infty$ に近づけるならば

$$\bigcup_{i=0}^{k} F([a_i, a_{i+1}]) \tag{11}$$

は $\{f(x)\mid x\in[a, b]\}$ に収束する. (11) の区間幅は $F([a, b])$ のそれより狭くなり, よりよい評価が得られる[4]. しかし計算機上での実現においては, k の値は演算時間との関係で制約を受けるため適当に定めなくてはならない.

以上が区間演算の区間幅拡大を防止する手段のすべてではないが, 有理区間関数を評価する上で有効である.

3. 多峰性関数の最適化への応用

n 次元ユークリッド空間のある領域において多峰性関数

$$y=f(x_1, x_2, \cdots\cdots, x_n) \qquad (n\geq 1) \tag{12}$$

が与えられているものとする. ここで (12) は1つ以上の極大値または極小値をもつと仮定する. このときの最大値を求めるアルゴリズムを次に示す.

<u>アルゴリズムA</u>

ここで目的関数および定義域を以下のように仮定する.

$$y=f(x_1, x_2, \cdots, x_n), \quad (a_1\leq x_1\leq b_1, \cdots\cdots, a_n\leq x_n\leq b_n) \tag{13}$$

<u>ステップ1</u>: $x_1, x_2, \cdots, x_m$, f を区間に拡張して $X_1=[a_1, b_1], \cdots\cdots, X_n=[a_n, b_n]$ と表し $F(X_1, \cdots, X_n)$ の区間数値を得る.

<u>ステップ2</u>: $X_1, X_2, \cdots X_n$ の区間幅の最大のものを分割する. もし区間 $[a_1, b_1]$ が最大ならば X_1 をその中点で分割し, $F([a_1, m], X_2, \cdots, X_n)$, $F([m, b_1], X_2, \cdots, X_n)$ として評価関数値 $[\underline{F_L}, \overline{F_L}]$, $[\underline{F_R}, \overline{F_R}]$ を得る. ここで $m=(a_1+b_1)/2$ は中点, $\overline{F_L}$ $(\overline{F_R})$, $\underline{F_L}$ $(\underline{F_R})$ は関数値の上(下)限を表す. いま図1のごとく $\overline{F_L}<\underline{F_R}$ ならば領域 $[a_1, m]\otimes\cdots\otimes X_n$ に最大値は存在しないので捨る. $\overline{F_R}<\underline{F_L}$ の場合も同様である. そうでなければ両者を保存する. 分割は, 区間関数値の上限が最大のものを対象とし, その最

大辺を中点で2等分する．
ステップ3：$X_1, X_2, \ldots, X_n$ および F の区間があらかじめ定めた範囲内になればステップ4へ，そうでなければステップ2を繰返し実行する．
ステップ4：いま ℓ 個の領域 $\varphi_i (i=1,2,\ldots,\ell)$ が残れば $\varphi = \bigcup_{i=1}^{\ell} \varphi_i$ を解とする．

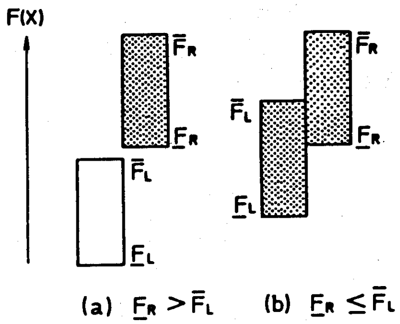

図1．関数値による取捨選択

本アルゴリズムにより関数の大域的な最大値を求めることが可能である．得られた解は，そこに種々の誤差を含めて評価したもので，解のとりうる範囲を誤差限界をも含めて正確に把握できる．これにより，多峰性関数をある定義域内において最適化できる．しかしこの計算過程において，関数の評価するとき2の区間幅拡大防止のアルゴリズムを取入れてもステップ3における φ は大きく解の範囲は狭まりにくい．そこで最適化向けの方法として次のアルゴリズムを追加した．図2において $[\underline{F_R}, \overline{F_R}]$ が求める最大値を含むとすると，$[\underline{F_L}, \overline{F_L}]$ の領域を捨てることができれば解としての区間は縮小される．そこで $[\underline{F_R}, \overline{F_R}]$ の領域内の1点（例えば中点 m）で $\underline{F_R}$ を置換することにより，もとの区間関数値の下限を引き上げられる．しかも解を含む領域を誤って捨てることも生じない．この結果，ステップ4における φ は十分小さくなり得られる解の区間は満足するものとなる．

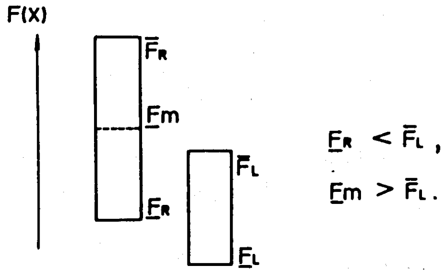

図2．中点による下限の引上

4．Newton 法と部分領域の統合

4．1 Newton の区間反復法[4)]

変数の定義域があまり広くなければ3のアルゴリズムで満足する結果が得られる．2の区間幅拡大を防ぐアルゴリズムを利用して計算時間を短縮できる．しかし定義域が広く，かつ／または，目的関数が“なべ底”であるときは，多くの部分領域が残り計算時間は変数の数が増加すると共に急速に増大する．

そこで一般の最適化の方法を取入れることが有効となる．ここでは，こうした一般の最適化の技法の一方法である Newton 法を区間解析用に拡張した手法を取入れることとした．

いま，1変数の実有理関数の極値を Newton 法を用いて求めることを考える．$f(x)$ を区間 $[a, b]$ で連続微分可能な実有理関数とする．$f(x)$ の極値を求めることは区間解析用に拡張した Newton 法において $f'(x) \equiv g(x) = 0$ の根を求めることになる．$g(x)$ の導関数 $g'(x)$ を有理区間関数に拡張したものを $G'(X)$ とする．ここで X の中点を $m(X)$ とおけば

$$N(X) = m(X) - \frac{g(m(X))}{G'(X)} \qquad \text{ただし}\ 0 \notin G'(X) \tag{14}$$

と定義される関数を用いることにより，Newton 法による区間列は

$$X_{p+1} = X_p \cap N(X_p), \qquad (p=0,1,\cdots) \tag{15}$$

となり $g(x)=0$ の根に収束する.
ここで $N(X_p)$ は $g(m(X))$ に区間関数を用いて

$$N(X_p) = m(X_p) - \frac{G(m(X_p))}{G'(X_p)} \qquad ただし\ 0 \notin G'(X_p) \tag{16}$$

となる.

n変数のとき, $f(x)$はn次元直方体 $A = A_1 \otimes A_2 \otimes \cdots \otimes A_n$ で連続微分可能であるとする. 多変数関数の最大値を求める問題は, 各変数について微分して0とおくことにより非線形連立方程式 $\nabla f(x) \equiv g(x) = 0$ の根を求める問題に帰着する.
ここで

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right), \quad x = (x_1, x_2, \cdots x_n) \tag{17}$$

である. $g(x)$のヤコビ行列は

$$J(x) = \left\{\frac{\partial g_i}{\partial x_j}\right\} \tag{18}$$

で与えられる. (16) に対応して

$$\mathbf{N}(\mathbf{X}_{(p)}) = m(\mathbf{X}_{(p)}) - J^{-1}(\mathbf{X}_{(p)}) \cdot G(m(\mathbf{X}_{(p)})) \tag{19}$$

となる. ここで $J^{-1}(\mathbf{X}_{(p)})$ は (18) を区間に拡張したものの逆行列である. これより $\mathbf{X}_0$ をとったとき (15) に対応して

$$\mathbf{X}_{(p+1)} = \mathbf{X}_{(p)} \cap \mathbf{N}(\mathbf{X}_{(p)}), \qquad (p=0,1,\cdots) \tag{20}$$

により $\mathbf{X}_{(p)}$は解に収束する.

なおここでJ^{-1}なる区間逆行列を求めるとき, 逆行列の計算公式に直接区間演算を適用するかわりに, 逆行列の計算値の誤差の評価に区間演算を適用することで逆行列を評価する Hansen の方法[5] を利用することができる.

ここで注意することは, (20) が成り立つのは, $\mathbf{X}_0$内に1つの極値が存在する場合に限定されることである.

4.2 部分領域の分離

評価関数 $f(x_1, x_2, \cdots, x_n)$が, 定義域内において関数値が同値な双峰の極値を持つような場合, 3のアルゴリズムでは共通部分をもたない変数の領域が複数個保存されるので, そのまま和集合をとって解とすることはできない. 当然4.1の Newton 法を適用することもできない. そこである程度3. のアルゴリズムで区間の分割を進めた後, 分割された領域がいくつかの共通点をもたない領域群に分離できれば, 群ごとに和集合をとって解とする. 変数の区間幅が十分狭くて領域内に単峰を有するのであれば, 4.1 の Newton法を適用できる. そこで領域の分離を以下のアルゴリズムで実現する.

<u>アルゴリズム B</u>

<u>ステップ1</u>: 3のアルゴリズムで順次変数の区間分割を進め, 複数個の部分領域を得る.

<u>ステップ2</u>: 1つの部分領域を取出しグループを作り, 残存領域群からグループ

内の領域と共通点をもてばそのグループに含める．

ステップ3：同じグループに1つでも入れば残された部分領域についてステップ2の後半を実行する．こうしてすべての部分領域から共通点をもつグループを1つの部分領域群として抽出分離する．

ステップ4：以上のプロセスの後に残された部分領域があればステップ2にもどる．そうでなければ，得られた部分群ごとに，各変数について和集合をとり，領域が十分小さければステップ5へ，そうでなければステップ1にもどり再度区間の分割を進めてこのプロセスを繰り返す．

図3．は数値実験例2の50回の区間分割の後 G_1，G_2 なる2つの部分群に分離した状況を示す．

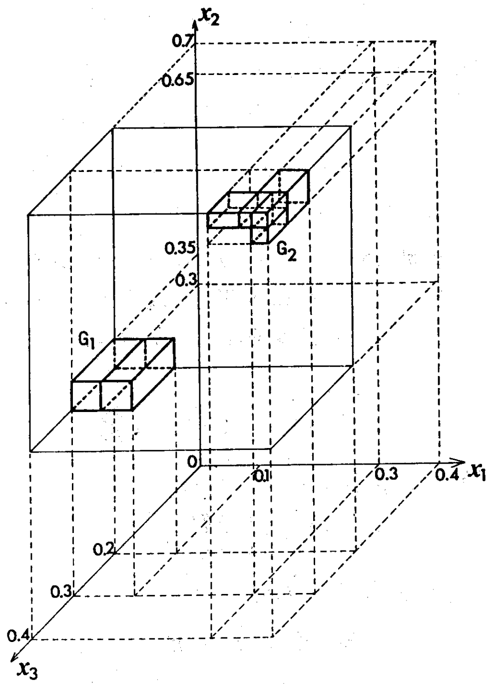

図3．部分群の分離と再統合

ステップ5：ステップ4によって各変数の和集合をとった結果，区間幅が収束判定条件を満たすなら解として区間関数の最大値（最小値）が求められる．そうでなければ，ここで得た領域をあらたな定義域として Newton法を適用して最適化する．もしこれでも収束しない場合はステップ1にもどり再度このプロセスを実行する．

5．領域選択と Newton法

既に述べたアルゴリズムにより，多峰性多変数関数の定義域内におけるすべての極値を求めることが可能である．ここでは4.1の Newton法を変数の区間分割の早い時期から適用し部分領域の分離をおこなうことなく極値を得ようとするもので，分割回数，目的関数の評価回数低減を目指した．

n 変数の場合の極値探索のアルゴリズムは次の通りである．

アルゴリズムC

ステップ1：各変数の定義域を区間として与える．

ステップ2：(18)の $|J| = det\ J(X)$ を区間として得る．

ステップ3：$0 \in |J|$ を調べて，$0 \notin |J|$ ならば，4.1で述べた Newton法を適用する．途中で収束判定条件に達した微小領域を解として保存する．そうでない場合（(20)の右辺が空集合となり，解が領域外に存在することを示す），解を含む可能性が無いので捨てる．

ステップ4：$0 \in |J|$ の場合は，$0 \in g(X)$ かどうか調べる．もし $0 \in g(X)$ ならば，解を含む可能性があるので，その領域を保存する．$0 \notin g(X)$ ならば解を含まないのでその領域を捨てる．

ステップ5：残存領域の内，変数区間幅最大のものを抽出しこれを2等分してステップ2へもどる．

以上の2～5のステップを繰返した後，保存していた最適解を取り出す．図4．は数値実験例4.の変数の区間分割と本アルゴリズムの適用により部分領域が捨てられて行く過程を示したものである．

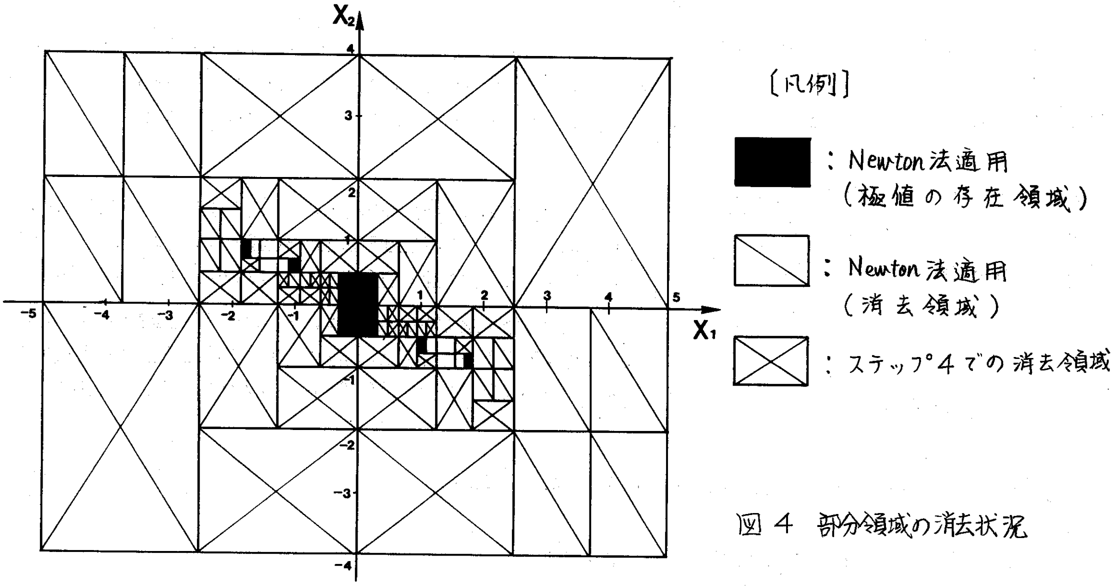

図４　部分領域の消去状況

6．数値実験例

例1．アルゴリズム A を次の関数(Rosenbrock Function)に適用した例[1]．

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (21)$$

定義域 $-1.2 \le x_1 \le 1.3$，$-1.4 \le x_2 \le 1.5$ として最大値，最小値を求めた．区間として得られた値は右の通りである．分割数は最大値で76回，最小値で262回である

X_1=[1.29999 99999 8, 1.30000 00000 0],
X_2=[-1.40000 00000 0, -1.39999 99999 9],
F_{max}=[954.89999 9976, 954.90000 0000],
X_1=[0.99999 99999 75, 1.00000 00003 0],
X_2=[0.99999 99998 37, 1.00000 00008 5],
F_{min}=[-0.13471 94D-16, 0.00000 00000 0].

例2．アルゴリズム A，及び B を適用した例，最大値が2ヶ所に存在する．

$$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3 - 1)^2 + \frac{1}{4}(x_2 - \frac{1}{2})^2 + \frac{1}{4}(x_3 - 0.3)^2 + 1 \quad (22)$$

定義域 $0.0 \le x_1 \le 0.4$，
$0.3 \le x_2 \le 0.7$，
$0.2 \le x_3 \le 0.4$．

として最大値，最小値を求めた なお右は288回の分割後の結果である．

X_1=[0.00000 00000 00, 0.28421 70943 04D-14],
X_2=[0.30000 00000 00, 0.30000 00000 00],
X_3=[0.20000 00000 00, 0.20000 00000 00],
F_{max}=[1.26250 00000 00, 1.26250 00000 00],
X_1=[0.40000 00000 00, 0.40000 00000 00],
X_2=[0.70000 00000 00, 0.70000 00000 00],
X_3=[0.40000 00000 00, 0.40000 00000 00],
F_{min}=[1.26250 00000 00, 1.26250 00000 00].

例3．アルゴリズムA，B 及びNewton法を適用した例．

i) 3-Hump Camel-Back Function.[6]

$$f(x_1, x_2) = a x_1^2 + b x_1^4 + c x_1^6 - x_1 x_2 + d x_2^2 + e x_2^4 \qquad (23)$$

$a=-2$, $b=1.05$, $c=-\frac{1}{6}$, $d=-1$, $e=1$.

定義域 $-2.0 \le x_1 \le 2.5$,
$-1.0 \le x_2 \le 1.5$.

正解は $x_1 = x_2 = 0$ のとき $f_{max}=0$ である．またこのとき右に示す2つの極大値が存在した．

X_1=[-0.14009 48867D-9, 0.41773 40292D-7],
X_2=[-0.83756 96620D-7, 0.62934 91594D-7],
F_{max}=[-0.10079 84362D-13, 0.67575 72253D-16].

X_1=[1.74755 23458 30288, 1.74755 23458 30289],
X_2=[-0.87377 61729 15144, -0.87377 61729 15144],
F_{max}=[-0.29863 84422 36869, -0.29863 84422 36853],

X_1=[-1.74755 23458 30295, -1.74755 23458 30283],
X_2=[0.87377 61729 15127, 0.87377 61729 15162],
F_{max}=[-0.29863 84422 36906, -0.29863 84422 36815].

ii) 5変数関数[2]

$$f(x_1, x_2, \cdots, x_5) = f_1(x_1) f_2(x_2) f_3(x_3) f_4(x_4) f_5(x_5), \qquad (24)$$

ここで

$$\left.\begin{aligned} f_1(x_1) &= x_1(x_1+13)(x_1-15)*0.01, \\ f_2(x_2) &= (x_2+15)(x_2+1)(x_2-8)*0.01, \\ f_3(x_3) &= (x_3+9)(x_3-2)(x_3-9)*0.01, \\ f_4(x_4) &= (x_4+11)(x_4+5)(x_4-9)*0.01, \\ f_5(x_5) &= (x_5+9)(x_5-9)(x_5-10)*0.01, \end{aligned}\right\} \qquad (25)$$

定義域 $-10 \le x_i \le 10$,
$(i=1, 2, \cdots, 5)$.

X_1=[8.75644 07330 07731, 8.75644 07330 07731],
X_2=[-9.35828 66332 94910, -9.35828 66332 94910],
X_3=[-4.57207 78818 33904, -4.57207 78818 33904],
X_4=[3.59212 96115 43726, 3.59212 96115 43726],
X_5=[-2.84008 63924 84075, -2.84008 63924 84045],
F_{max}=[24416.03065 50573 6, 24416.03065 50573 8].

例4．アルゴリズムCを(23)に適用した例．

定義域 $-5 \le x_1 \le 5$，$-4 \le x_2 \le 4$．

下に示す結果①は最大値，②は極値，③は鞍点を示す．なお②，③については，それぞれ原点対称の極値及び鞍点が存在する．

①
X_1=[-0.81908 08599 94908D-18, 0.81902 79204 35705D-18],
X_2=[-0.76427 78283 07555D-18, 0.76427 78283 07555D-17],
$F(\mathbf{x})$=[-0.59753 84679 46713D-34, 0.62600 53408 85193D-35],

②
X_1=[1.74755 23458 30288, 1.74755 23458 30288],
X_2=[-0.87377 61729 15144, -0.87377 61729 15144],
$F(\mathbf{x})$=[-0.29863 84422 36886, -0.29863 84422 36831],

③
X_1=[1.07054 22918 23659, 1.07054 22918 23661],
X_2=[-0.53527 11459 11831, -0.53527 11459 11829],
$F(\mathbf{x})$=[-0.87736 15577 63152, -0.87736 15577 63129].

以上の数値実験例の内，最小値は $f \leftarrow -f$ としてもとめた．なお，計算結果に付した下線は上下限（左：下限，右:上限）の一致を示している．

7．おわりに

区間解析による多峰性多変数関数の極値探索の方法について述べた．ここでは有理関数の定義域内における大域的な最大値（最小値）ならびに極値をもとめたが，変数の数，峰の数が増えるに従って計算時間は急速に増加する．しかし，5のアルゴリズムCによる極値探索法は計算時間も比較的短かく，数値実験例4で計算時間約1.4秒（HITAC M-200Hを使用）であり十分多変数の計算に用いることができるので，今後引き続き数値実験を行ってみたい．

一方，既に述べたアルゴリズムによって得られる解は，高精度であるので他の極値探索の数値実験結果を比較する場合にも利用できると思われる．

参考文献

[1] Dixon,L.C.W.,Szegö,G.P.:Towards global optimization.Amsterdam: North Holland (1975).

[2] 澤田，佐藤：多峰性多変数関数の最大・最小，情報処理，Vol. 16，No.1，pp. 2-6 (1975).

[3] Bremermann,H.:A method of unconstrained global optimization. Mathematical Bioscience,Vol,9,pp.1-15(1970).

[4] Moore,R.E.:Interval analysis.New York: Prentice-Hall(1966).

[5] Hansen,Eldon:Topics in interval analysis.Oxford Univ. Press(1969).

[6] Branin,F.H.:Widely convergent of simultaneous nonlinear equations.IBM J. Vol.9,pp.504-522(1972).