



区間演算を利用した多変数関数の最大値探索法*

藤井康雄** 市田浩三*** 清野 武***

Abstract

A method is described for finding maxima and minima of multivariable functions. Using interval arithmetic, one can find the global maximum or minimum with rigorous error bounds. The convergence can be made fast by Newton's method after subregions are grouped.

1. まえがき

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、Variable Metric法など数多くの方法¹⁾が発表されており、また多変数になるとランダムサンプリング法、指数関数による relative peaking と多重積分を組合せた方法²⁾などが有効であると言われている。さらに Bremerman 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⁾。

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は、常にその大域的な最大、最小が求められる方法は現在のところない。決定論的な方法では、いくつかの出発点から極値を求めるアルゴリズムを繰り返して大域的な解を推定するのが普通である。確率的方法では、大域的解に到達する可能性があるがきわめて多数回の試行が必要である。

ここでは、区間解析の手法を多峰性多変数関数に適用し、ある領域内での大域的な最大、最小を求める方法について述べる。関数を区間関数に拡張し、領域を順次分割して各部分領域での関数値の上下限を推定し、最適解を含む可能性のない領域を捨てていくことによってその最大値を求めるアルゴリズムを提案す

る、これは単純なアルゴリズムであるが、極めて正確に最大値、または最小値を求めることができる。また、各部分領域の集りを、それぞれ独立した部分群として分類し、その後通常最適化技法の Newton 法を区間関数用に拡張した方法を適用して最適解を求める方法についても述べる。

区間解析は、今日までにさまざまな方面へ利用されている⁴⁾⁻⁶⁾。しかし区間解析は、その特質を演算時における誤差評価として持っているため、数値解析の誤差解析の一環として発達してきた感が強い。実際に区間演算を実施するのに作成されたシステムとしては、triplex⁷⁾というタイプの区間の算術的結合への拡張を含む1つの Algol コンパイラ 'Triplex-Algol-Karl-sruhe' があるにすぎない。他には、Algol またはその他の言語で区間演算を実行する一部分が作られている程度である^{8),9)}。多くのコンパイラでは、区間演算用のステートメントがまったく用意されていないのが実情で、もし用意するとすれば機械語の知識が必要となるので、個々の計算機に対して別々に作成しなければならない。将来、多くのコンパイラにおいて区間演算への拡張を含むようになるか、あるいはもっとフレキシブルな言語が使用できるようになれば区間演算もさらに利用されることと思われる。ここで使用した区間演算システムは、FORTRAN とアセンブラで書かれ、任意の桁での切り上げ、切り捨てができるシステムとして構成されている。

2. 区間演算

2.1 区間演算の定義^{4),10)}

いま任意の数を、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わし、すべての数値を幅

* A Method for Finding the Greatest Value of a Multivariable Function Using Interval Arithmetic by Yasuo FUJII (Department of Electronics, Faculty of Engineering, Kyoto University), Kozo ICHIDA and Takeshi KIYONO (Department of Information Science, Faculty of Engineering, Kyoto University)

** 京都大学工学部電子工学科

*** 京都大学工学部情報工学科

をもった数値として取り扱い、これを「区間」と呼ぶ、

$$[a, b] = \{x | a \leq x \leq b\} \quad (1)$$

とくに $[a, a] = a$

とするとき、これら区間の間の基本演算を次のように定義する。

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

ただし $0 \notin [c, d]$.

(2)の区間演算のうち、加法と乗法については、ともに結合法則と交換法則を満たす。もし、 I, J および K を区間とすれば、以下の関係が成立する。

$$\left. \begin{aligned} I + (J + K) &= (I + J) + K, \\ I \cdot (J \cdot K) &= (I \cdot J) \cdot K, \\ I + J &= J + I, \\ I \cdot J &= J \cdot I \end{aligned} \right\} (3)$$

しかし、区間演算の場合、分配法則は必ずしも成立しない。分配法則の代わりに区間 I, J, K に対し

$$I \cdot (J + K) \subset I \cdot J + I \cdot K \quad (4)$$

が成り立つ。もし、 J と K が同符号の実数だけを含むときは分配法則

$$I \cdot (J + K) = I \cdot J + I \cdot K \quad (5)$$

が成り立つ。したがって区間演算は、包含単調である。つまり $I \subset K, J \subset L$ ならば

$$\left. \begin{aligned} I + J \subset K + L, \quad I - J \subset K - L, \quad I \cdot J \subset K \cdot L, \\ I / J \subset K / L, \quad \text{ただし } 0 \notin L \end{aligned} \right\} (6)$$

が成立する。(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。しかし、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、区間幅拡大を防ぐ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

すでに述べた区間の四則演算で関数値を評価する場合を考える。通常、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。このため以下に述べるよ

* k は多変数になれば、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。

うな方法によって拡大を防ぐのが一般である。

i) 入れ子形 (nested form)

多項式

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0, \quad a_i \in A_i, \quad x \in X \quad (7)$$

(ただし A_i, X はそれぞれ a_i, x の区間)

の評価を考える。(7)を区間関数にして区間演算として求めるならば

$$F(X) = A_n X^n + A_{n-1} X^{n-1} + \dots + A_0 \quad (8)$$

によるが

$$F(X) = (\dots((A_n X + A_{n-1})X + \dots + A_1)X + A_0 \quad (9)$$

の‘入れ子’にしたほうが演算の結果(8)より区間幅が狭くなる⁴⁾。

ii) 中心形 (centerd form)

$f(x_1, x_2, \dots, x_n)$ を実有理関数とし、実数 $c = (c_1, c_2, \dots, c_n)$ を f の値が定義されるベクトルで

$$f(x_1, x_2, \dots, x_n) = f(c_1, c_2, \dots, c_n) + g(x_1 - c_1, \dots, x_n - c_n) \quad (10)$$

と表わせるとする。区間 $[a, b]$ の中点は、 $m([a, b]) = (a+b)/2$ で表わされることより、区間 X_i の中点を $m(X_i)$ で表わすとき

$$F(X_1, X_2, \dots, X_n) = f(m(X_1), m(X_2), \dots, m(X_n)) + G(X_1 - m(X_1), \dots, X_n - m(X_n)) \quad (11)$$

となり $F(X_1, X_2, \dots, X_n)$ 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) 自乗演算のときの処理

$A \cdot A = \{B | B \in [c, d]\}$ 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。これは、区間 A が 0 をはさんで正負の領域をとっている場合には i), ii) の方法より有効である。

iv) 区間の細分化

いま X の区間が $[a, b]$ で与えられるとき、

$$[a, b] = [a, a_1] \cup [a_1, a_2] \cup \dots \cup [a_n, b] \quad (12)$$

のように区間を細分化して考える。これは、 X を $(k+1)$ 等分したものであり*、 f を実関数とすると

$$\{f(x) | x \in [a, b]\} = \bigcup_{i=0}^k \{f(x) | x \in [a_i, a_{i+1}]\} \quad (13)$$

$(a_0 = a, a_{k+1} = b)$

である。有理区間関数は

$$\bigcup_{i=0}^k F([a_i, a_{i+1}]) \supset \{f(x) | x \in [a, b]\} \quad (14)$$

と表わせる。いま $k \rightarrow \infty$ に近づけるならば

$$\bigcup_{i=0}^k F([a_i, a_{i+1}]) \quad (15)$$

は $\{f(x)|x \in [a, b]\}$ に収束し、その区間幅は $F([a, b])$ のそれより狭い。(15)によってよりよい評価が得られる⁴⁾。

これらが、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

$$y = f(x_1, x_2, \dots, x_n) \quad \text{ただし } n \geq 1 \quad (16)$$

が与えられているとする。(16)は1つ以上の極大値または極小値をもつものとする。いま $n=2$ として関数の最大値を求めることを考える。このアルゴリズムは以下ようになる。ただし、関数および定義域を次のように仮定する。

$$y = f(x_1, x_2), (a \leq x_1 \leq b, c \leq x_2 \leq d). \quad (17)$$

ステップ 1: x_1, x_2 をそれぞれ区間に拡張して $X_1 = [a, b], X_2 = [c, d]$ と表わし、 $F([a, b], [c, d])$ の区間関数値を得る。ここで F は f の区間関数である。

ステップ 2: X_1 と X_2 の区間の幅の大きい方を分割する。いま $(b-a) > (d-c)$ であったとすると X_1 をその中点 $m = (a+b)/2$ で分割し、 $(X_{1L} = [a, m], X_2)$ と $(X_{1R} = [m, b], X_2)$ の2組から $F([a, m], [c, d]), F([m, b], [c, d])$ としてそれぞれの区間関数を評価して、区間関数値 $[E_L, \bar{F}_L], [E_R, \bar{F}_R]$ を得る。ここで $\bar{F}_L, \bar{F}_R, E_L, E_R$ は、それぞれ関数値の上(下)限を表わすものとする。もし $\bar{F}_L < E_R$ であれば領域 $[X_{1L}, X_2]$ に最大値が存在しないことは明らかであるから、この領域は捨てることができる。 $\bar{F}_R < E_L$ のときも同様である。そうでないときは両方の領域を残す (Fig. 1)。

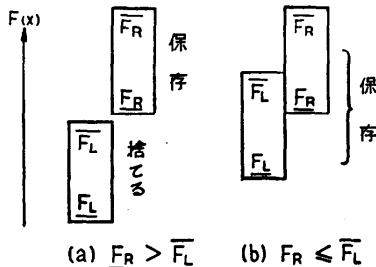


Fig. 1 Interval value of $F(X)$ in two subregions
(a) one of the subregion can be discarded
(b) both subregions must be retained

ステップ 3: 以下各部分領域を順次分割していくが、分割する領域は区間関数値の上限が大きいもので、最も区間幅の大きな変数の区間をその中点で2等分する。こうして得られた関数値から解を含む可能性のある領域を保存する。

ステップ 4: X_1 および X_2 の区間、または関数値の区間があらかじめ定めた範囲内になればステップ 5 へいく。そうでないならステップ 3 へもどる。

ステップ 5: いま l 個の領域 $\phi_i (i=1, 2, \dots, l)$ が残ったとすれば $\bigcup_{i=1}^l \phi_i$ を解とする。

以上のようにして、このアルゴリズムによって関数の大域的な最大値を求めることができる。 $n > 2$ の場合についても同様である。また最小値を求める場合は、 $f \leftarrow -f$ として求めればよい。

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あり、解のとらえる範囲を誤差限界をも含めて正確に把握することができる。このようにして多峰性関数を、ある領域内において最適化することができる。このプロセスにおいて区間関数の計算のとき 2.2 で述べた区間幅拡大防止のアルゴリズムを適用しなければよい評価は得られない。しかしながら区間演算を最適問題に適用するには、これだけでは演算時間が長くなること、ステップ 5 において l が大きくなり解の区間が狭まりにくい、などの理由で実用性に乏しくなる。そこで最適化向けに考慮しなければならない。いま Fig. 2 で $[E_R, \bar{F}_R]$ が求める最大値を含むとすると $[E_L, \bar{F}_L]$ の領域を捨てれば解の区間は縮小されることになる。それには、 E_R をできる限り \bar{F}_R に近づけて区間を狭めるのが最適である。このため $[E_R, \bar{F}_R]$ の領域内の1点 (たとえば中点 m) で求めた区間関数値の下限で E_R を置き換えることにより、もとの区間関数値の下限を引き上げることが可能であり、しかも解を含む領域を誤って捨て去ることもおこらない。

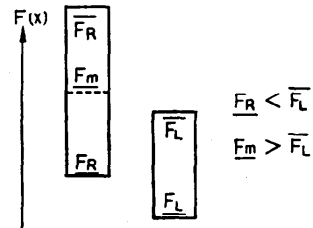


Fig. 2 Pulling up the lower bound of interval value of $F(X)$

すなわち $f(x_1, x_2)$ の最大値は, $[F_m, \bar{F}_R]$ の中に存在する.

この結果, ステップ5における領域 φ は, 少なくとも得られる解の区間は充分満足するものとなる.

4. Newton 法と領域の分離

4.1 Newton の区間反復法⁴⁾

各変数の定義域が狭い場合には, 3. のアルゴリズムで充分満足な関数の評価が可能である (計算例1を参照). その場合, 評価すべき関数の性質から区間幅拡大を防ぐアルゴリズムを適宜選択して利用しなければならない. ことに多変数になるほど, 実変数が何回も現われる有理式も多く, これを変形して各変数がただ1回だけ現われるように書換えることは不可能となる. このため区間拡大防止のアルゴリズムの2.2のうち iii), iv) で述べたアルゴリズムを主として使用するのが有効となる. しかし, 関数の性質が, 逆向きの 'なべ底' のような場合の最大値を求めるときには, 3. のアルゴリズムでは, 解の区間幅が拡がりやすく, 演算時間も長くなるので一般の最適化の方法を利用することが有効となる. ここでは最適化の一方法である区間演算に拡張された Newton 法について述べる.

まず, 1変数の実有理関数の場合 $f(x)=0$ の根を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し, その微係数 $f'(x)$ を有理区間関数に拡張したものを $F'(X)$ とする. ここで X の中点を $m(X)$ とおけば,

$$N(X) = m(X) + \left(\frac{1}{-F'(X)} \right) f(m(X)) \quad (18)$$

ただし $0 \notin F'(X)$

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

$$X_{p+1} = N(X_p) \cap X_p, \quad (p=0, 1, 2, \dots) \quad (19)$$

となり $f(x)=0$ の根に収束する.

つぎに, n 次元の場合について考える. 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 $f(x)=0$ の根を求める問題に帰着する. ここで $x=(x_1, x_2, \dots, x_n)$, $f(x)=(f_1(x), f_2(x), \dots, f_n(x))$ である. $f(x)$ は n 次元直方体, $A=A_1 \otimes A_2 \otimes \dots \otimes A_n$ で連続微分可能であるとする. $f(x)$ のヤコビ行列 $J(x)$ は

$$J(x) = \begin{Bmatrix} \frac{\partial f_1}{\partial x_j} \\ \vdots \\ \frac{\partial f_n}{\partial x_j} \end{Bmatrix} \quad (20)$$

である. (18) は $f(m(X))$ に区間関数を用いて

$$N(X) = m(X) - \frac{F(m(X))}{F'(X)} \quad \left. \vphantom{N(X)} \right\} \quad (21)$$

ただし $0 \notin F'(X)$

と書き換えられる. 多次元の場合には

$$N(X_{(p)}) = m(X_{(p)}) - J^{-1}(X_{(p)}) \cdot F(m(X_{(p)})) \quad (22)$$

となる. ここで $J^{-1}(X_{(p)})$ は (20) を区間に拡張したものの逆行列である. これより適当な X_0 をとったとき

$$X_{(p+1)} = X_{(p)} \cap N(X_{(p)}) \quad (p=0, 1, 2, \dots) \quad (23)$$

によって $X_{(p)}$ は解に収束する. ただし $x \in X_0$ に対し行列 $J(x)$ は正則でなければならない. これより多変数の場合も, 1変数のときと同様にして求められる.

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¹¹⁾ を利用することができる.

ここで注意することは, Newton 法を適用する場合, 区間内に必ず1つの極値を持っている場合に限定される. さらに, 変数の区間 $[a, b]$ とするとき, その端 (a , または b) に最適解が存在する場合には, Newton 法を適用できない. この場合, 3. のアルゴリズムで解を求めればよい.

4.2 領域の分離

評価する関数 $f(x_1, \dots, x_n)$ が, その定義域内において多峰性であっても同値, あるいはほとんど同値な最大値を持つような場合, 3. で述べたアルゴリズムでは, 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不適當である. 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あと, 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, 群ごとに各変数について和集合をとって解とする. 変数の区間幅が充分狭くて領域内に単峰を有するのであれば 4.1 で述べた 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.

ステップ1: 3. のアルゴリズムで順次区間の分割を進め, いくつかの部分領域を得る.

ステップ2: 1つの部分領域を取り出しグループを作る. 残っている部分領域群から, 順次1つの部分領域を取り出しグループ内の部分領域と共通点をもてばグループに入れる.

ステップ3: 同じグループに1つでも入れれば, 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.

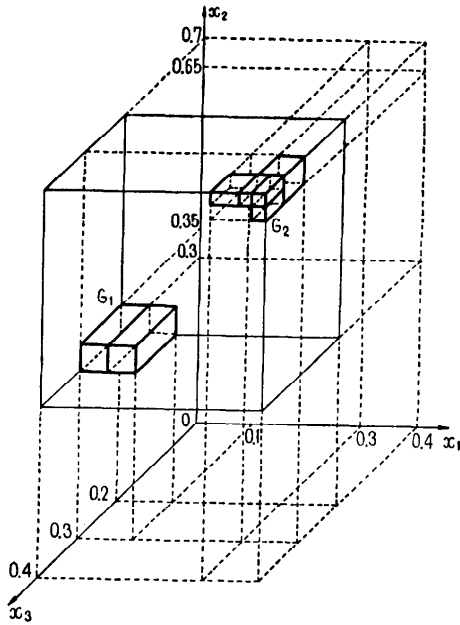


Fig. 3 Graph of the situation of example 2 after the original region is divided 50 times and the remaining subregions are grouped

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、得られた各部分群ごとに、各変数について和集合をとり、領域が充分小さければステップ 5 へ、そうでないならばステップ 1 にもどり再び区間の分割を進めてこのプロセスを繰り返す。Fig. 3 は、計算例 2 の演算の途中で部分領域が 2 つの部分群 G_1 , G_2 に分離された状況を示したものである。

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足するならば解として区間関数の最大値(最小値)が求められる(計算例 2 参照, ただしこれは 4.1 で注意した Newton 法が適用できない例である)。区間幅が解とするには広すぎるときには、ここで得たそれぞれの領域を、あらたな定義域として Newton 法を適用して最適化する。もしこれでも最適解を得られなければ、ステップ 1 にもどって再度このプロセスを実行する。なお、上のアルゴリズムを区間分割のたびに実施して Newton 法を適用すれば、ある領域内における、すべての極大値、または極小値を見出すことも可能である(計算例

3 の i) 参照)。

5. 不規則領域の場合の最適化

関数の領域が不規則な形をしているときは、3. で述べたアルゴリズムをそのまま適用することはできない。ただし特別な場合には、以下に述べる変換法によって不規則領域を多次元の矩形領域に変換して最適解を求めることができる^{12), 13)}。

5.1 2 変数の場合

まず、2 変数の場合を考える。条件

$$g(x_1, x_2) \leq 0 \tag{24}$$

のもとで関数 $f(x_1, x_2)$ の最大値を求めることを考える。(24) で表わされる領域は単連結で、その境界

$$r = u(\theta) \tag{25}$$

は、1 価関数であると仮定する (Fig. 4)。ここで

$$r = (x_1^2 + x_2^2)^{1/2}, \theta = \tan^{-1}(x_2/x_1) \tag{26}$$

である。次の変換

$$x_1 = tu(\theta) \cos \theta, x_2 = tu(\theta) \sin \theta \tag{27}$$

によって、問題は、条件

$$0 \leq t \leq 1, 0 \leq \theta \leq 2\pi \tag{28}$$

のもとで $f(tu(\theta) \cos \theta, tu(\theta) \sin \theta)$ の最大値を求めることに帰着される。領域 (28) は矩形領域であることから (Fig. 5), 3 の方法がそのまま利用できる。条件 (24) が複数個

$$\left. \begin{aligned} g_1(x_1, x_2) &\leq 0, \\ \dots\dots\dots \\ g_m(x_1, x_2) &\leq 0, \end{aligned} \right\} \tag{29}$$

あるときも、その境界が区分的に

$$\left. \begin{aligned} \theta_1 \leq \theta \leq \theta_2 \text{ のとき } r &= u_1(\theta), \\ \dots\dots\dots \\ \theta_k \leq \theta \leq \theta_{k+1} \text{ のとき } r &= u_k(\theta), \end{aligned} \right\} \tag{30}$$

と表わせるときは*, やはり矩形領域において最適解を求めることに帰着できる(計算例 4 参照)。

5.2 多変数の場合

n 変数のときは、その領域

$$g(x_1, x_2, \dots, x_n) \leq 0 \tag{31}$$

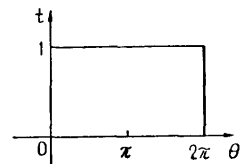
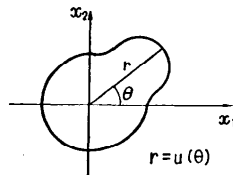


Fig. 4 An irregular region Fig. 5 A transformed rectangular region

* (29), (30) において一般に $k \geq m$.

の境界が

$$r = u(\theta_1, \theta_2, \dots, \theta_{n-1}) \quad (32)$$

と表わされたとする、変換

$$\left. \begin{aligned} x_1 &= tu \sin \theta_1 \sin \theta_2 \dots \sin \theta_{n-1}, \\ x_2 &= tu \sin \theta_1 \sin \theta_2 \dots \cos \theta_{n-1}, \\ &\dots\dots\dots \\ x_{n-1} &= tu \sin \theta_1 \cos \theta_2, \\ x_n &= tu \cos \theta_1 \end{aligned} \right\} \quad (33)$$

によって、問題は条件

$$\left. \begin{aligned} 0 \leq t \leq 1, \\ 0 \leq \theta_i \leq \pi, \quad (i=1, 2, \dots, n-2) \\ 0 \leq \theta_{n-1} \leq 2\pi \end{aligned} \right\} \quad (34)$$

のもとで

$$\begin{aligned} f(tu \sin \theta_1 \dots \sin \theta_{n-1}, tu \sin \theta_1 \dots \cos \theta_{n-1}, \\ \dots, tu \sin \theta_1 \cos \theta_2, tu \cos \theta_1) \end{aligned} \quad (35)$$

の最大値を求めることになる。(35)は矩形領域であるから2変数の場合と同様に、3.のアルゴリズムをそのまま利用することができる。

6. 計算例

例1 3.のアルゴリズムを次の関数(Rosenbrock Function)に適用した例。

$$f(x_1, x_2) = a(x_2 - x_1^2)^2 + (b - x_1)^2 \quad (36)$$

定義域 $-1.2 \leq x_1 \leq 1.3, -1.4 \leq x_2 \leq 1.5$
 $a=100, b=1$ として最大値, 最小値を求めた。区間として得られた値は次の通りである。

$$\begin{aligned} X_1 &= [1.29999\ 99999\ 8, 1.30000\ 00000\ 0], \\ X_2 &= [-1.40000\ 00000\ 0, -1.39999\ 99999\ 9], \\ F_{max} &= [954.89999\ 9976, 954.90000\ 0000] \end{aligned}$$

分割数 76.

$$\begin{aligned} X_1 &= [0.99999\ 99999\ 75, 1.00000\ 00003\ 0], \\ X_2 &= [0.99999\ 99998\ 37, 1.00000\ 00008\ 5], \\ F_{min} &= [-0.13471\ 94 \times 10^{-16}, 0.00000\ 00000\ 0] \end{aligned}$$

分割数 262.

例2 3., 4.2のアルゴリズムを適用した例, 最大値が2ヶ所に存在する。

$$\begin{aligned} f(x_1, x_2, x_3) &= (x_1 + x_2 + x_3 + a)^2 + b(x_2 + c)^2 \\ &+ b(x_3 + d)^2 + e \end{aligned} \quad (37)$$

$$a = -1, b = 1/4, c = -1/2, d = -0.3, e = 1$$

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

$$X_1 = [0.00000\ 00000\ 00, 0.28421\ 70943\ 04 \times 10^{-14}],$$

$$X_2 = [0.30000\ 00000\ 00, 0.30000\ 00000\ 00],$$

$$X_3 = [0.20000\ 00000\ 00, 0.20000\ 00000\ 00],$$

$$F_{max} = [1.26250\ 00000\ 00, 1.26250\ 00000\ 00].$$

$$X_1 = [0.40000\ 00000\ 00, 0.40000\ 00000\ 00],$$

$$X_2 = [0.70000\ 00000\ 00, 0.70000\ 00000\ 00],$$

$$X_3 = [0.40000\ 00000\ 00, 0.40000\ 00000\ 00],$$

$$F_{max} = [1.26250\ 00000\ 00, 1.26250\ 00000\ 00]$$

分割数 288,

例3 3., 4.のアルゴリズムを適用した例。

i) 3-Hump Camel-Back Function

$$f(x_1, x_2) = ax_1^2 + bx_1^4 + cx_1^6 - x_1x_2 + dx_2^2 + ex_2^4 \quad (38)$$

$$a = -2, b = 1.05, c = -1/6, d = -1, e = 0$$

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

$$X_1 = [-0.14009\ 48867 \times 10^{-9}, 0.41773\ 40292 \times 10^{-7}],$$

$$X_2 = [-0.83756\ 96620 \times 10^{-7}, 0.62934\ 91594 \times 10^{-7}],$$

$$F_{max} = [-0.10079\ 84362 \times 10^{-13}, 0.67575\ 72253 \times 10^{-16}].$$

これ以上 Newton 法を適用すれば、指数部がアンダーフローを起こす。なお正解は $x_1 = x_2 = 0$ のとき $f_{max} = 0$ である。またこのとき次の2組の極大値が存在した。

$$X_1 = [1.74755\ 23458\ 30288, 1.74755\ 23458\ 30289],$$

$$X_2 = [-0.87377\ 61729\ 15144, -0.87377\ 61729\ 15144],$$

$$F_{max} = [-0.29863\ 84422\ 36869, -0.29863\ 84422\ 36853],$$

$$X_1 = [-1.74755\ 23458\ 30295, -1.74755\ 23458\ 30283],$$

$$X_2 = [-0.87377\ 61729\ 15127, 0.87377\ 61729\ 15162],$$

$$F_{max} = [-0.29863\ 84422\ 36906, -0.29863\ 84422\ 36815].$$

ii) 6-Hump Camel-Back Function

(38)において、 $a = -4, b = 2.1, c = -1/3, d = 4, e = -4$ とおいたもの、最大値は2ヶ所に存在する。

定義域 $-2.5 \leq x_1 \leq 2.0, -1.5 \leq x_2 \leq 2.0$

$$X_1 = [-0.08984\ 20131\ 00318\ 08, -0.08984\ 20131\ 00318\ 05],$$

$$X_2 = [0.71265\ 64030\ 20739\ 7, 0.71265\ 64030\ 20739\ 7],$$

$$F_{max} = [1.03162\ 84534\ 89877, 1.03162\ 84534\ 89878].$$

$$X_1 = [0.08984 \ 20131 \ 00318 \ 05, \ 0.08984 \\ 20131 \ 00318 \ 08],$$

$$X_2 = [-0.71265 \ 64030 \ 20739 \ 7, \ -0.71265 \\ 64030 \ 20739 \ 7],$$

$$F_{\max} = [1.03162 \ 84534 \ 89877, \ 1.03162 \\ 84534 \ 89878].$$

i), ii) とともに文献 3), 14) と比較して, 極めて良い結果である. なお文献 14) では, 小数点以下 3 桁までの解が与えられているが, その範囲では上の結果と一致している.

例 4 5. のアルゴリズムを適用した例

$$\left. \begin{aligned} f(x_1, x_2) &= (x_1 - 1)^2 + x_2^2 \\ \text{ただし } (x_1^2/4) + x_2^2 &\leq 1 \end{aligned} \right\} \quad (39)$$

の最大値を求める. これは (28) のもとで

$$f(t, \theta) = \left(\frac{2t \cos \theta}{\sqrt{1+3 \sin^2 \theta}} - 1 \right)^2 + \left(\frac{2t \sin \theta}{\sqrt{1+3 \sin^2 \theta}} \right)^2 \quad (40)$$

の最大値を求めることになる.

$$T = [0.99999 \ 97615, \ 1.00000 \ 00000],$$

$$\theta = [3.14159 \ 26535, \ 3.14159 \ 30280],$$

$$F_{\max} = [8.99999 \ 85694, \ 9.00000 \ 00000] \text{ 分割数 } 340.$$

7. むすび

区間演算を利用して, 多峰性多変数関数を最適化する方法について述べた. ここでは有理関数を主体とし, 3 変数, 6 峰までのそれぞれの定義域内における大域的な最大値 (または最小値) を求めたが, より多峰性, 多変数の場合には, 区間関数の評価において, 関数値の拡がり, 関数値の下限の引き上げなどをいま以上にできる方法を考えなければならない. さらに区間演算システムの充実と処理速度の改善など, 今後検討する予定である.

こうした精度の高い最適解は, 他の最適化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) J. Kowalik, & M. R. Osborne (山本, 小山共訳): 非線形最適化問題, 培風館 (1971).
- 2) 津田, 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) H. Bremermann: A Method of Unconstrained Global Optimization, Mathematical Biosciences, Vol. 9, pp. 1~15 (1970).
- 4) R. E. Moore: Interval Analysis, Prentice-Hall, Series in Automatic Computation, New Jersey (1966).
- 5) E. Hansen: Topics in Interval Analysis, Oxford University Press (1969).
- 6) 町田, 市田, 清野: 区間解析による二点境界値問題の解, 情報処理, Vol. 16, No. 6, pp. 552~554 (1975).
- 7) K. Nickel & others: The Algorithmic Language Triplex- ALGOL 60, Numer. Math., Bd. 11, pp. 175~180 (1968).
- 8) H. Christ: Realisierung einer Maschinen-Intervallarithmetik mit Beliebigen Algol-60-Compilern, Electron. Rechenanl, Jg. 10, H. 5, S. 217~222 (1968).
- 9) T. D. Ladner, J. M. Yohe: An Interval Arithmetic Package for the UNIVAC 1108, Mathematics Research Center in the Univ. of Wisconsin (1970).
- 10) 室谷義昭: インタバル・アナリシス, bit 臨時増刊号, pp. 1303~1311 (1975).
- 11) E. Hansen, R. R. Smith: A Computer Program for Solving a System of Linear Equation and Matrix Inversion with Automatic Error Bound Using Interval Arithmetic, LMSC 4-22-66-3, Lockheed Missiles & Space Co., California (1966).
- 12) T. S. Chow, H. W. Milnes: Solution of Laplace's Equation by Boundary Contraction over Regions of Irregular Shape, Numer. Math., Bd. 4, pp. 209~225 (1962).
- 13) K. Ichida, T. Kiyono: Numerical Integration in the Irregular Region, Computing, Vol. 12, pp. 9~15 (1974).
- 14) F. H. Branin, Jr.: Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations, IBM J. Vol. 9, pp. 504~522 (1972).
- 15) 藤井, 市田, 清野: 区間解析を利用した多峰性関数の最適化, 情報処理学会 16 回大会講演論文集, pp. 629~630 (1975).

(昭和 51 年 6 月 7 日受付)