



多峰性多変数関数の最大・最小と図形会話型処理*

津田 孝夫** 佐藤 美枝子**
大久保 英嗣*** 吉田 秀逸***

Abstract

This paper discusses a new algorithm to locate the global maximum of a function defined in a multi-dimensional rectangular domain. The number of dimensions is as large as, or even more than, 5 or 10. There are two important elements in this algorithm. One is the transformation of the object function in such a way that its global maximum corresponds to infinity while other secondary maxima are reduced to zero. Actually there is some departure from the ideal transformation because of possible overflows on computer. This portion of the algorithm precedes the interactive (or conversational) use of a graphic display system. This interactive part makes the other element of the algorithm. A multi-dimensional point is represented as a curve on the display screen. By projecting numerous points in the multi-dimensional space to similarly numerous curves on the screen of the graphic display device, the human eye can make overall recognition much more efficiently than computers. This fact is exploited to reduce the problem to that of a set of uni-modal peaks. Once the supporting domain for each of these peaks is separated by visual aid, one may leave the computer to handle the rest of the problem for itself. A number of numerical experiments are done and discussed to evidence the feasibility of the proposed algorithm.

1. 序 論

本論文は、1975年夏の第2回日米コンピュータ会議で発表した著者らの論文¹⁾の続編であって、その後の新しい展開を論ずることを目的とする。そこでまず、前回までの要点を記し、問題のありかと、それに対し著者らが新しく提案しようとしている方法論の志向するところを、この序論でまとめておこう。多次元の問題は人間の直観が伴わないことが普通で、むしろ機械処理に適しているように想像されがちであろう。しかし、ここでは、多次元の情報に適切に図形表示装置の

画面上に集約的に写像して表現し、このような2次元図形のパターンに対し人間の判断力を活用して、マン・マシンを一体として考える広義のアルゴリズムが多変数問題の処理にきわめて有効であることを例証しようとするものである。

目的は、 k 次元ユークリッド空間 ($k \gg 1$) の有限領域で定義された多峰性多変数関数の大域的な最大値 (global maximum) を求めることにある。最小値の場合も同工異曲であるので、ここではとくにふれない。

一般に、いままでにこの問題に対し提案されてきた方法は、すべて局所的 (local) な方法であって、探索の出発点によっては局所的な極大に捕捉されてしまうのである。あるいは、探索領域に一樣にランダムに順次点列をとり、それらの点における関数値を評価し、比較することにより、一番大きな値を与える点を大域的な最大値を与える点とする、というのであるが、これ

* Interactive Search for the Global Maximum of a Function of Very Many Variables by Takao TSUDA, Mieko SATO (Faculty of Engineering, Hokkaido University), Eiji OKUBO and Shuitsu YOSHIDA (Graduate School of Hokkaido University)

** 北海道大学工学部電気工学科

*** 北海道大学大学院工学研究科情報工学専攻

も次元数が高い場合には、すでに報告したように¹⁾、まことに効率(いわば解像力)が悪い。以上要するに、いままでは局所的な方法か、ランダム・サンプリングか、ということになるのであるが、これに対し著者らは、計算機とそれに結合された図形処理装置を用いた新しい大域的アルゴリズムを提案しようというのである。

本アルゴリズムは2段に分かれており、前段はほぼ前回報告した通りの内容である¹⁾。後段は、それを受けて、大型機と結合した端末の図形処理装置を通して行う会話型処理(これを図形会話型処理と呼ぼう)であって、著者らは北海道大学大型計算機センターのFACOM 230-75をホスト・コンピュータとして、工学部内のTSS 端末GT-40(DEC社製の図形処理装置で、CPUはPDP 11/10)を用いた。なおその際、著者らのグループで開発した図形表示用ソフトウェア・システムIGDFS(Interactive Graphic Display FORTRAN System)²⁾は、はなはだ有効であった。

上述のアルゴリズムの前段とは次のようなものであって、計算機の自動処理に適した部分である。

与えられた関数

$$F(x) = F(x_1, x_2, \dots, x_k) \quad (1)$$

の大域的最大値を単位超立方体

$$D = D_1 \times D_2 \times \dots \times D_k, \quad (2)$$

$$D_i \triangleq \{x_i | 0 \leq x_i \leq 1\} \quad (i=1, 2, \dots, k) \quad (3)$$

で見出した。そこであらたに

$$f(x) \triangleq \exp[F(x)] \quad (4)$$

を定義する。ここで $0 \leq F(x) < \infty$ で、しかも $F(x)$ の最大値は D 内のただ1点でしか起こらないと仮定し、その点を $x = x_M$ とする。このとき

$$\frac{[f(x)]^n}{\int_D [f(x)]^n dx} \rightarrow \delta(x - x_M), \quad (n \rightarrow \infty) \quad (6)$$

が示せるので ($\delta(x)$ は Dirac のデルタ関数)、十分大きな正の定数 c (ただし計算の各段でオーバーフローを起こさせない程度に大きい数) を用いて

$$\phi(x) \triangleq \exp[c(F(x) - F_0)] \quad (6)$$

を考えると、

$$\bar{x} \triangleq \int_D x \phi(x) dx / \int_D \phi(x) dx \quad (7)$$

が求めたい x_M の1つの近似になっていることは、デルタ関数の性質から明らかであろう ($c \rightarrow \infty$ ならば \bar{x} は正確であるが、 $c \rightarrow \infty$ とできないための問題が生ずるのである)。ただし(6)の F_0 は予備計算の段階で

求まる $F(x)$ の最小値 F_{\min} である。この予備計算¹⁾とは、少数のサンプル点について $F(x)$ をあらかじめ調べることににより、上述のように原関数をデルタ関数に“整形”するために必要な定数を定める前処理的な過程である。 c や F_0 は正確に求めることは要しない。このように、指数関数の性質を巧妙に使って、大域的な最大値を計算機で扱える無限大まで釣り上げることで、それをもとに多重積分法³⁾を駆使して計算を実行することが、アルゴリズムの前段の中心になっている。

(7)式を評価すると同時に、解のちらばりを示す2次のモーメント

$$\sigma_i^2 = \int_D (x_i - \bar{x}_i)^2 \phi(x) dx / \int_D \phi(x) dx \quad (i=1, 2, \dots, k), \quad (8)$$

$$\begin{aligned} \sigma^2 &= \int_D (x - \bar{x})^2 \phi(x) dx / \int_D \phi(x) dx \\ &= \sigma_1^2 + \sigma_2^2 + \dots + \sigma_k^2 \end{aligned} \quad (9)$$

の値を求める。ここでもし、(7)式で求めた推定値 \bar{x} の近傍に解 x_M が存在するという保証が一旦得られたら、次に探索領域を、その推定値 $x = \bar{x}$ を中心とした小さな矩形領域に縮小し、さらに探索する。この区間縮小をくり返すことによって、解の改善をはかる。

予備計算¹⁾で見出された $F(x)$ の最大値 F_{\max} に対応する

$$\phi_i \triangleq \exp[c(F_{\max} - F_0)] \quad (10)$$

を閾値(threshold)と呼ぼう。(7)、(8)の多重積分の評価に必要な N 個のサンプル点のうち、 $\phi(x) \geq \phi_i$ となる点の数を N_i (≥ 1) とし、一方 $\phi(x) \geq \phi_i$ となる D 内の k 次元体積を D_i とすると、

$$D_i / D \cong N_i / N \quad (11)$$

が成り立つ。

一方、 $x = \bar{x}$ を中心とした半径 3σ の k 次元球の体積 V_0 は、

$$V_0 \begin{cases} = (2\pi)^{k/2} (3\sigma)^k / k!! & (k = \text{偶数}) \\ = 2(2\pi)^{(k-1)/2} (3\sigma)^k / k!! & (k = \text{奇数}) \end{cases} \quad (12)$$

で与えられる。そこで、order of magnitude において

$$D_i / D \cong V_0 \quad (13)$$

が成立するならば、領域 D_i は単連結で、それが近似的に球状領域をなしているための十分条件が成立したとみなしうるので、最大値をとる点 x_M が球 V_0 の中に捕捉されたとして、区間縮小に移ることができる。

しかし、

$$D_i / D \ll V_0 \quad (14)$$

ならば、 D_t は Fig. 1 に概念図で示したように、単連結球状領域でなく、同程度の峰が2つ以上あったり、あるいはたとえば、非常に細長い場合に対応する。

$$D_t/D \gg V_v \tag{15}$$

のときは、Fig. 2 の概念図に見られるように、 $\phi(x)$ は台 D_t の上部において微細構造をもち、 V_v の大きさはその微細構造の“幅”によって定まる。この場合は閾値 ϕ_t が低すぎるのであって、それをさらに高くする必要がある。詳しくは文献1) 参照のこと。

以上のように、2つの独立な量 D_t/D と V_v の相互の大小関係により、(13)、(14)、(15)の3つの場合が生ずるが、(15)の場合は最終的には(13)か(14)に帰着する。ところで、数値実験を重ねた結果、(14)の状態が終わる事例がかなり多く、その場合いままでのアルゴリズム¹⁾では、推定値 \bar{x} は正しいと保証できないという警告を出力して、計算を打ち切った。しかしこのような場合が多いのでは、判断は安全側であるとはいえ、問題を解決したことにはならないので、(14)の場合をさらに詳しく吟味する必要が起こるのである。

2. グラフィック・ディスプレイ装置による図形会話型処理

要するに問題となるのは、多次元空間内の点集合 D_t の構造 (たとえば Fig. 1 参照) を知ることであ

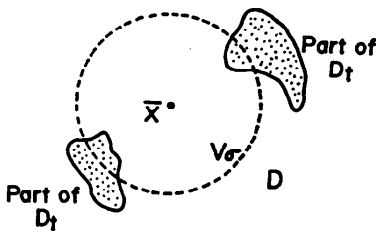


Fig. 1 Case (14): $D_t/D \ll V_v$. V_v is the k -dimensional volume of the sphere (shown by dotted line) with its center at estimate \bar{x} .

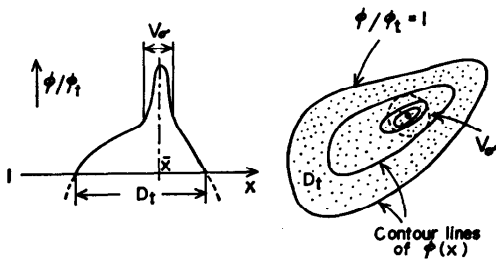


Fig. 2 Case of (15): $D_t/D \gg V_v$.

る。 D_t に属するサンプル点がいくつかのクラスター (cluster) を形成していれば、1つのクラスターが1つの (単) 峰に対応していると考えられ、それらのクラスターごとに最大値を求めて比較するという簡単な問題になる。

ここで、 k 次元空間の1点 $x=(x_1, x_2, \dots, x_k)$ を

$$y=y(x, t) = \frac{1}{\sqrt{2}}x_1 + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + x_5 \cos 2t + \dots, \quad -\pi < t \leq \pi \tag{16}$$

なる1本の2次元曲線に対応させてみよう。この変換によって、点 x をグラフィック・ディスプレイ装置のスクリーン上に表わすことができる。ここで、 t はスクリーン上の横軸に対する変数で、関数 y の値を縦軸にとれば、点 x に対応する1本の曲線が連続関数としてスクリーン上に描かれる⁴⁾。

(16)はまた、次のような2つのベクトルの内積とも考えられる。すなわち、

$$y = x \cdot e(t). \tag{17}$$

ここで $e(t)$ は

$$e(t) = (1/\sqrt{2}, \sin t, \cos t, \sin 2t, \cos 2t, \dots) \tag{18}$$

で定義されるベクトルで、次元数が奇数ならば正確に一定長、また次元数 k が偶数ならばほぼ一定長 ($k \gg 1$ とする) であるベクトルである。

領域 $D_t \triangleq \{x | \phi(x) \geq \phi_t, \text{ in } D\}$ から有限個のサンプル点をぬき出したとき、同じクラスターに属する任意の2点は、 k 次元空間で同じ共通の近傍内にある。したがって、 k 次元空間から2次元空間への写像(16)が、その変換の前後において距離を保存していれば、スクリーンの画面上で人間の視覚によって、複数個の異なるクラスターの存在を弁別できるであろう。実際、この距離の保存関係は次のようにして成り立つ。

k 次元空間の2点 x, x' 間のユークリッド距離 $d(x, x')$ に対し、画面上における2曲線 $y(x, t), y(x', t)$ の間の“ずれ”は、視覚的には2乗平均の意味で

$$\delta(x, x') \triangleq \left[\int_{-\pi}^{\pi} \{y(x, t) - y(x', t)\}^2 dt \right]^{1/2} \tag{19}$$

のように測られよう。一方、

$$d(x, x') = \left[\sum_{i=1}^k (x_i - x'_i)^2 \right]^{1/2} \tag{20}$$

であるので、

$$\delta(x, x') = \sqrt{\pi} d(x, x') \tag{21}$$

が成り立つ。ゆえに、変換(16)を行っても、距離は保存される。

さらに $\langle \mathbf{X} \rangle$ を, N 個の点 $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$ の平均

$$\langle \mathbf{X} \rangle = \frac{1}{N} \sum_{j=1}^N \mathbf{X}_j \quad (22)$$

とすると,

$$y(\langle \mathbf{X} \rangle, t) = \frac{1}{N} \sum_{j=1}^N y(\mathbf{X}_j, t) \quad (23)$$

が成り立つ。ゆえに, k 次元空間内の複数個の点の平均は, 画面上では曲線群の平均として描かれる。

また, (21)式から次のことがいえる。同一のクラスターに属する点は, ディスプレー装置の画面上でもある近さをもった曲線群となるため, 1本の帯状の曲線群を形作る。 k 次元空間内にクラスターが2つあれば, 画面上には2つの曲線群の帯となるので, 視覚的に見分けられることができる。

本論文の目的のために必然的に多くの曲線を同時に見るわけであるが, グラフィック・ディスプレイではそれぞれの曲線の identity を簡単に確めることができるという重要な機能がある。プロッター出力のような静的な図形では, 近接しながら交差している一本一本の曲線の identity を確めることは容易でない。これに比し, グラフィック・ディスプレイでは, 各曲線にあらかじめラベルをつけておいて, この呼び名によってキ・ボードから任意の曲線を指定したり, またはライト・ペンで直接画面に現われている曲線の1つを指定して, それを十分に高い輝度でブリンク (blink) させることにより, 各曲線の identity を確めることができる。また, 1つのクラスターを作っていると思われる曲線群だけを画面に表示し, 他の曲線を消去することも簡単に行える。

また, スクリーン上で同時に表わせる点の数は, せいぜい 10 点か 20 点であって, これ以上になると曲線が重なりすぎて見分けにくい。そのため, D_i 内に 20 個以上の点がある場合には, それらから適当に重なりを許しながら 10 点ないし 20 点をくり返しとり出し表示することにより, D_i の構造を調べることができる。

3. アルゴリズム

以上述べてきたアルゴリズムの前段と後段をとりまとめると, 次のようになる。

- 1) 一様乱数を用いて探索領域 D 内に N_0 個 (比較的少数個) のサンプル点を取り, それらの点における関数値 $F(\mathbf{x})$ を計算して, その最大と最小 F_{\max}, F_{\min} を求め, c, F_0, ϕ_i の値を定める。

- 2) (7), (8) 式の積分を計算する。(9)は(8)から出る。積分のサンプル数 N に関し, N_i がある程度の数 (たとえば $N_i=10$) になるまで, この N を増やしながら積分の評価を続ける。次に D_i/D ($\equiv N_i/N$) と V_i の値を求める。

- 3) (13), (14), (15)のいずれの場合が起こったか, チェックする。(13)の場合は 5)へ, (14)の場合は 6)へ, (15)の場合は 4)へ進む。

- 4) N_0 を増やして (たとえば 2 倍にして) 1)へ戻る。すなわち ϕ_i を高くして再計算する。

5)

- a) すべての $i(=1, 2, \dots, k)$ に対し, (8)式の σ_i が σ_c (あらかじめ定めてある目標精度) 以下なら計算を終了し, そうでないときは区間縮小を行う。区間縮小は, 変数 x_i の前回の区間を $[a_i, b_i]$ とするとき, 新しい区間を

$$\{x_i \mid \max\{a_i, \bar{x}_i - (b_i - a_i)\beta/2\} \leq x_i \leq \min\{b_i, \bar{x}_i + (b_i - a_i)\beta/2\}\}$$
 とする (たとえば, $\beta=1/2$ にとる)。

- b) この新しい区間に対して 1)を行い, サンプル数 N で(7), (8)の積分を計算して, 再び 5)のはじめに戻る。

6)

- a) $\phi(\mathbf{x}) \geq \phi_i$ となる k 次元空間の領域 D_i 内の N_i 個の点を, (16)式により N_i 本の曲線として画面上に表示する。各曲線には自動的にラベル (番号) がつけられるが, この番号と画面上の曲線の対応関係や, 1本の曲線のつながり具合を交錯する他の曲線から区別して確かめたいときには, 当該曲線の番号をキ・ボードから入力して, その曲線を高い輝度でブリンクさせる。

- b) N_i 本の曲線を視覚によりいくつかのクラスターに分け, 各クラスターにどの曲線が属しているか, 計算機に教える。各クラスターごとに点の座標の平均 $(\text{XBAR})_i$ と分散 $(\text{VAR})_i$ を各座標軸方向について求める ($i=1, 2, \dots, k$)。

- c) 1つのクラスターに関し, x_i の新しい探索区間を

$$\{x_i \mid (\text{XBAR})_i - 3(\text{VAR})_i^{1/2} \leq x_i \leq (\text{XBAR})_i + 3(\text{VAR})_i^{1/2}\}$$

(ただし, 前回の区間よりはみ出す部分があれば, トリム (trim) する) として, 1)へ戻る。このとき, 3)のチェックでまた(14)の場合がお

こったときは、そのことを出力して再びくり返す。

- d) c)をb)で発見された異なるクラスターすべてにくり返した後、それぞれの結果を比較し、その中で最大のものを解とする。

4. 数値実験

次の関数について計算を行った。

$$F(x_1, x_2, \dots, x_5) = F_1(x)F_2(x_2)\dots F_5(x_5),$$

$$F_1(x_1) = x_1(x_1+13)(x_1-15) \times 0.01,$$

$$F_2(x_2) = (x_2+15)(x_2+1)(x_2-8) \times 0.01,$$

$$F_3(x_3) = (x_3+9)(x_3-2)(x_3-9) \times 0.01,$$

$$F_4(x_4) = (x_4+11)(x_4+5)(x_4-9) \times 0.01,$$

$$F_5(x_5) = (x_5+9)(x_5-9)(x_5-10) \times 0.01.$$

(1) 探索区間をまず、

$$-10 \leq x_i \leq 10 \quad (i=1, 2, \dots, 5)$$

としてみよう。 $N_0=1000, N=5000, N_i=10, \beta=1/2, \sigma_e=0.01$ とする。結果を Table 1 に示す。この場合、3)の判定で $D_i/D \cong V_e$ [(13)] であって、われわれのプログラムでは図形表示を要求しないが、一応念のため、 D_i 内の 10 点を曲線として、別にプロッターで描かせてみた。それを Fig. 3 に示す。たしかに D_i は単峰関数の台となっていることがこれからもわかる。

Table 1 Computed results for $-10 \leq x_i \leq 10$ ($i=1, 2, \dots, 5$) with $N_0=1000, N=5000$.

preprocessing							
trial	number of sample points		threshold	D_i/D	V_e		
	N	N_i					
1st	7696	10	0.1505×10^6	0.1299×10^{-3}	0.4687×10^{-3}	$D_i/D \gg V_e$	
2nd	30266	10	0.1780×10^6	0.3304×10^{-3}	0.6574×10^{-3}	$D_i/D \cong V_e$	
interval contraction							
	number of interval contraction	x_1	x_2	x_3	x_4	x_5	value of $F(x)$
true solution		8.7564	-9.3582	-4.5720	3.5921	-2.8400	24416.03
computed results	7	8.7599	-9.3611	-4.5708	3.5929	-2.8430	24416.01

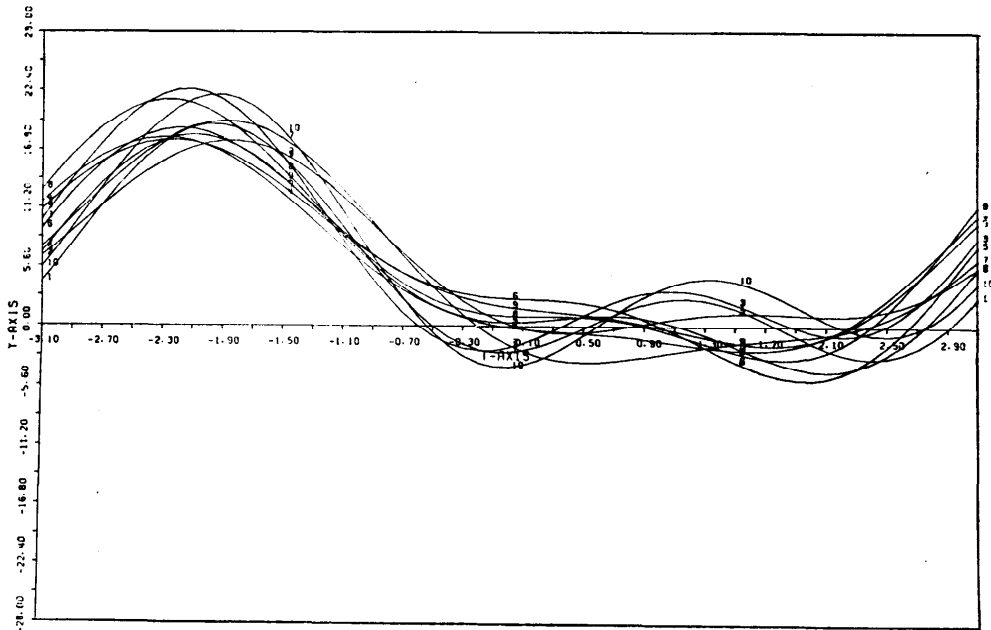


Fig. 3 Plotter output for $-10 \leq x_i \leq 10$ ($i=1, 2, \dots, 5$). All of 10 points of D_i , shown as 10 curves, form a single cluster.

Table 2 Computed results for $-10 \leq x_1 \leq 8$, $-10 \leq x_2 \leq 11$, $-10 \leq x_i \leq 10$ ($i=3, 4, 5$) with $N_0=500$, $N=5000$.

(a)

preprocessing						
trial	number of sample points		threshold	D_t/D	V_e	
	N	N_t				
1st	5000	36	0.7538×10^4	0.7200×10^{-2}	0.9224×10^{-15}	$D_t/D \gg V_e$
2nd	12030	10	0.1302×10^5	0.8312×10^{-3}	0.6358×10^{-5}	$D_t/D \gg V_e$
3rd	40219	10	0.1754×10^5	0.2486×10^{-3}	0.3424×10^{-1}	$D_t/D \ll V_e$

(b) Interactive search (step 6)

[Cluster 1]

preprocessing						
trial	number of sample points		threshold	D_t/D	V_e	
	N	N_t				
1st	10814	10	0.2539×10^5	0.9247×10^{-3}	0.1252×10^{-3}	$D_t/D \approx V_e$

interval contraction

	number of interval contraction	x_1	x_2	x_3	x_4	x_5	value of $F(x)$
true solution		8.0	11.0	-4.5720	3.5921	-2.8400	27604.21
computed results	7	7.9998	10.999	-4.5715	3.5914	-2.8456	27604.03*

[Cluster 2]

preprocessing						
trial	number of sample points		threshold	D_t/D	V_e	
	N	N_t				
1st	5000	16	0.2225×10^5	0.3200×10^{-3}	0.2687×10^{-4}	$D_t/D \gg V_e$
2nd	17180	10	0.2317×10^5	0.5820×10^{-3}	0.3915×10^{-4}	$D_t/D \gg V_e$
3rd	46122	10	0.2335×10^5	0.2168×10^{-3}	0.4599×10^{-4}	$D_t/D \approx V_e$

interval contraction

	number of interval contraction	x_1	x_2	x_3	x_4	x_5	value of $F(x)$
true solution		8.0	11.0	-4.5720	3.5921	-2.8400	27604.21
computed results	7	7.9999	-9.3590	-4.5713	3.5914	-2.8419	24139.81**

* $F_{max,1} = 27604.03$ ** $F_{max,2} = 24139.81$

(2) 次に探索区間を

$$-10 \leq x_1 \leq 8, \quad -10 \leq x_2 \leq 11, \quad -10 \leq x_i \leq 10$$

($i=3, 4, 5$)

としてみよう。 $N_0=500$, $N=5000$, $N_t=10$, $\beta=1/2$, $\sigma_e=0.01$ とする。結果を **Table 2** に示すが、同表 (a) からわかるように、 $D_t/D \ll V_e$ の判定が生じたためにグラフィック・ディスプレイのルーチンが稼動され、画面上に 10 本の曲線が描かれる。そのときの画面の様様を **Fig. 4(a)** (次頁参照) に示すが、同図 (b) はラベル 1 つのついた曲線をプリンクさせたときの画面の (ある瞬間の) 様子である。曲線 {1, 6, 7, 10} がクラスター 1 を、曲線 {2, 3, 4, 5, 8, 9} がクラスター 2 を形成していることが視覚で判定できる。そこで各クラスターについて計算を行うと、各クラスターは単

峰の台 ($D_t/D \approx V_e$) であって、結果が **Table 2 (b)** のように得られる。このとき、 $F_{max,1} > F_{max,2}$ となっているので、クラスター 1 の方から正解が得られる。

(3) 探索区間を

$$-10 \leq x_1 \leq 8, \quad -10 \leq x_2 \leq 12, \quad -10 \leq x_i \leq 10$$

($i=3, 4, 5$)

としてみよう。 $N_0=1000$, $N=5000$, $N_t=10$, $\beta=1/2$, $\sigma_e=0.01$ とする。結果を **Table 3** (次頁参照) に示す。この場合、 $D_t/D \approx V_e$ となるので、とくに画像は扱わない。

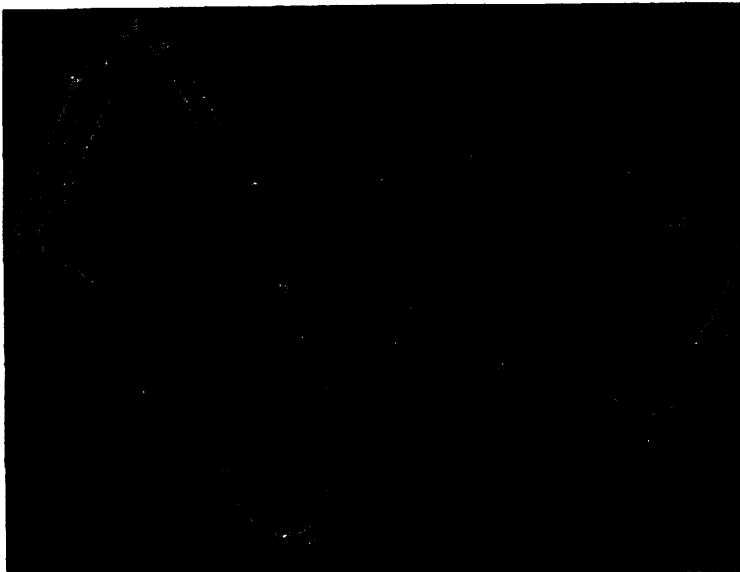
以上の他、いろいろな例について数値実験を行った。場合によっては、もちろんクラスターが 3 個生ずる場合も起こるが、そのような場合でも各曲線の高低

(819 頁につづく)

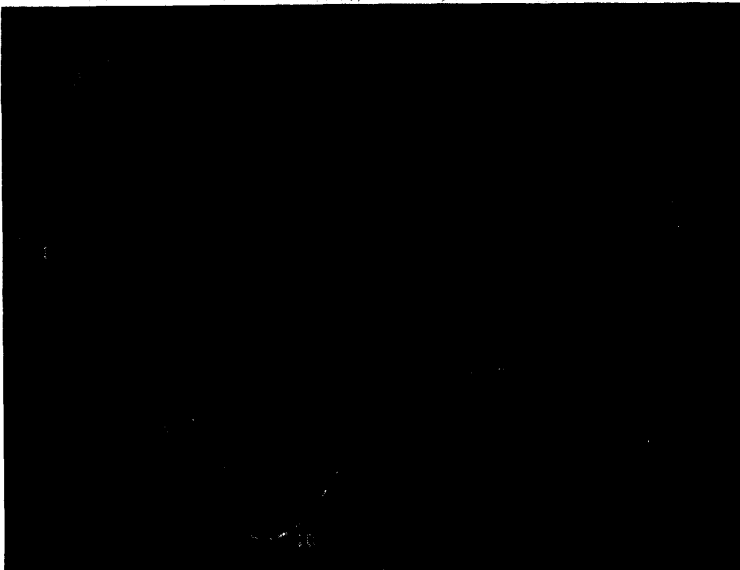
Table 3 Computed results for $-10 \leq x_1 \leq 8$, $-10 \leq x_2 \leq 12$, $-10 \leq x_i \leq 10$ ($i=3, 4, 5$) with $N_0=1000, N=5000$

preprocessing						
trial	number of sample points		threshold	D_t/D	V_e	
	N	N_t				
1st	58751	10	0.2158×10^8	0.1702×10^{-3}	0.5316×10^{-8}	$D_t/D \gg V_e$
2nd	245623	10	0.2673×10^8	0.4071×10^{-4}	0.5482×10^{-4}	$D_t/D = V_e$

interval contraction							
	number of interval contraction	x_1	x_2	x_3	x_4	x_5	value of $F(x)$
true solution		8.0	12.0	-4.5720	3.5921	-2.8400	41406.32
computed results	7	7.9996	11.999	-4.5701	3.5884	-2.8322	41405.11



(a) 10 points of D_t are seen to form 2 clusters.



(b) Blinking the curve labeled as 1 for identification. This curve is specially shown as a broken line with high brightness.

Fig. 4 Graphic display for $-10 \leq x_1 \leq 8$, $-10 \leq x_2 \leq 11$, $-10 \leq x_i \leq 10$ ($i=3, 4, 5$).

部の分布や位相関係の大づかみな傾向を視覚的に把握して、適当なクラスターに分けた上、正解に到達することができた。

また、パラメータ N_0 と N は、それぞれ、閾値 ϕ の高低および積分の精度に関係するが、 N_0 を 500 と 1000 の場合に分けて調べても、やはり視覚的には同様な曲線群が得られる。また $N=2500$ と 5000 の場合を比較しても、結果には直接影響がないことも確かめた。

5. 結 論

以上要するに、多峰性多変数関数の最大・最小を与える点を見出すため、図形会話型処理を用いることが本質的に重要であることを論じた。

再言するまでもないが、プロッターのような出力機器は本論文の目的には適合しない。プロッター出力はあくまでも静的なものであって、これをもとに柔軟なオンライン的判断を人間はすることができない。また、入りまじった曲線群の各曲線を区間の端から端まで画面上で追って、その identity を確かめることも容易でない。このように、図形会話型処理の目的には、グラフィック・ディスプレイ装置が必要不可欠である。

多変数関数の最大・最小問題は、直接的には、最適化問題との関係において重要なことはいうまでもなく、非線形パラメータ推定や、多変数関数の最良近似等の基本にもなるので、今後、本論文で述べた手法がさらにそれらの方面で改良され、発展することを深く

希望する。

本研究は、津田の原案をもとに、大久保、吉田が図形会話型処理プログラムを作成し、北大情報工学専攻の端末装置 GT-40 を用いて行ったもので、北海道大学大型計算機センターの FACOM 230-75 をホスト・コンピュータとして利用させて頂いた。また、アルゴリズム前段は、主として佐藤がコーディングを行い、一方、プロッタ出力をグラフィック・ディスプレイ出力と比較して、計算過程の正しいことを確かめた。さらに、IGDFS (本文参照) の稼動にあたっては、木村洋司君に大変お世話になった。記して謝意を表する次第である。

参 考 文 献

- 1) T. Tsuda, Y. Tanaka and M. Sato: Numerical Method for Functions of Very Many Variables: Search for the Global Maxima with *A Posteriori* Error-Bound Estimation, Proceedings of the Second USA-Japan Computer Conference (Tokyo), pp. 83~87 (Aug. 1975).
- 2) 平山正治, 津田孝夫: 会話型図形表示システム, 北海道大学大型計算機センター・ニュース, Vol. 7, No. 6, pp. 25~53 (1975年12月).
- 3) T. Tsuda: [Numerical Integration of Functions of Very Many Variables, *Numerische Math.*, Vol. 20, pp. 377~391 (1973).
- 4) D.F. Andrews: Plots of High-Dimensional Data, *Biometrics*, Vol. 28, pp. 125~136 (1972).
(昭和51年1月9日受付)
(昭和51年2月23日再受付)