

JAERI-M  
7229

少変数の非線形最適化問題に対する一手法

1977年8月

鈴木 忠 和

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

少変数の非線形最適化問題に対する一手法

日本原子力研究所東海研究所原子炉工学部

鈴木 忠 知

(1977年7月27日受理)

少変数の非線形最適化問題を解く新しい手法が開発された。この手法の特徴は球面上に作られたいくつかの点における目的関数の値を評価し、最小値を与える点と球の中心を結ぶ方向上に球を移動させながら最適点を探索していく繰り返し法で、テスト関数として用いたRosenbrockの関数やBealeの関数のような等高線が狭く曲がっていたり深い谷を持つ問題に対しては効果的である。この新しい手法は一つのサブルーチンとして最適化コードシステムSCOOPの中に組みこまれた。

A New Method For Nonlinear Optimization  
Problems With A Few Variables

Tadakazu SUZUKI

Division of Reactor Engineering, Tokai Research  
Establishment, JAERI

(Received July 27, 1977)

A new method has been developed for solving nonlinear optimization problem with a few variables. It is an iterative method of searching the optimum point on a spherical surface in each iteration process by evaluating values of the objective function at several given points on the sphere and then moving the searching sphere in a direction giving the minimum objective.

The method is useful for the problems represented with the contours having narrow curves or deep valleys, such as the Rosenbrock's function and Beale's function which are test functions in the present report.

The new approach has been introduced in optimization code system SCOOP.

Keywords:

Nonlinear Optimization, A Few Variables, Spherical Search, Iterative Method, Benchmark Test, Rosenbrock's Function, Beale's Function, Optimization Code System SCOOP.

## 目 次

1. はじめに	1
2. アルゴリズムの概要	1
2.1 探索点とその空間座標	2
2.2 探索のアルゴリズム	3
3. テスト問題と計算結果	7
3.1 テスト問題	7
3.2 計算の条件	8
3.3 計算結果	8
3.4 収束効率の比較と検討	10
4. 結 論	25
謝 辞	25
参考文献	25

## Contents

1. Introduction.....	1
2. Algorithms of Spherical Search Method.....	1
2.1 Searching points and their co-ordinates.....	2
2.2 Searching procedure.....	3
3. Benchmark Tests.....	7
3.1 test problems.....	7
3.2 Description of test calculations.....	8
3.3 Numerical results.....	8
3.4 Computation efficiencies and discussions.....	10
4. Conclusions.....	25
Acknowledgment.....	25
References.....	25

## 1. はじめに

非線形最適化問題の難しさは、目的関数の変数の数の多少よりもその形にあることはいうまでもないことで、それ故多種多様な方法が提案されているがいずれも一長一短があり、いわゆるオール・マイティーな解法はないようである。例えば傾斜法の中で最も効率の良い最急降下法は、良く知られているように変数の小さな摂動に対して不安定であり、このことは最大傾斜の方向が良いという仮定そのものに起因している。これに対しFletcherとReeves<sup>[1]</sup>が提唱したのが共役方向を用いる方法であり共役傾斜法と呼ばれている。さらに計量行列を各反復毎に任意に選ぶという考え方から可変計量法がDavidon<sup>[2]</sup>により提唱されFletcherとPowell<sup>[3]</sup>により発展された。然しながらこの方法でもその精度が一次元探索の精度に左右されることが多い。HookeとJeeves<sup>[4]</sup>が提唱したPattern探索法は目的関数が持つ深い谷を見つけて進んで行く方法であるが最急降下法と同じ欠点を持っていることが知られている。Rosenbrock<sup>[5]</sup>はHookeとJeevesの方法を発展させた座標回転法を提唱し、さらにNelderとMead<sup>[6]</sup>はSimplex法を提唱しているが、それらの計算効率がその中で使われる定数の定め方に左右されるにも拘わらずそれらの定め方に基準がないという欠点をもっている。

我々がここで提案する方法は考え方としてはSimplex法に類似したもので、Simplex法が変数が作る空間でのSimplexの端点における目的関数の値を評価し、Simplexの折り返し、縮小、拡大という基本操作によりこれを最適点へと動かして行くのに対し、我々の方法はその空間における超球上での関数値を評価しこれを最適点へと動かして行くものである。

なお、この研究はシステム解析手法研究グループの活動の一環としてなされたもので、この新しい手法は一つのサブルーチンとして最適化コードシステムSCOOPの中に組みこまれた。

## 2. アルゴリズムの概要

ここで述べる解法は、与えられた球面上にできるだけ一様になるように探索点を作り、それらの点における目的関数の値を評価し、最小値（以後最適化問題とは目的関数の最小値を求める問題と考えることにする）を与える方向へ球を移動させながら最適点を探索していく繰り返し法で、球面探索法と呼ぶことにする。

アルゴリズムの説明の前に以下の量が既知であると仮定する。

$F(x)$  : 変数  $x \in E^n$  の関数 (目的関数)。

$x_0$  : 変数  $x$  の初期値。球の最初の中心として  $x_0$  がとられる。

$r_0$  : 最初の球の半径。

$N$  : 一球面上の探索点の数。

$\alpha$  : 球の半径縮小係数。 ( $0 < \alpha < 1$ )

$\beta$  : 球を移動させるときの距離。 ( $\beta > 1$ )

## 1. は じ め に

非線形最適化問題の難しさは、目的関数の変数の数の多少よりもその形にあることはいうまでもないことで、それ故多種多様な方法が提案されているがいずれも一長一短があり、いわゆるオール・マイティーな解法はないようである。例えば傾斜法の中で最も効率の良い最急降下法は、良く知られているように変数の小さな摂動に対して不安定であり、このことは最大傾斜の方向が良いという仮定そのものに起因している。これに対しFletcherとReeves<sup>[1]</sup>が提唱したのが共役方向を用いる方法であり共役傾斜法と呼ばれている。さらに計量行列を各反復毎に任意に選ぶという考え方から可変計量法がDavidon<sup>[2]</sup>により提唱されFletcherとPowell<sup>[3]</sup>により発展された。然しながらこの方法でもその精度が一次元探索の精度に左右されることが多い。HookeとJeeves<sup>[4]</sup>が提唱したPattern 探索法は目的関数が持つ深い谷を見つけて進んで行く方法であるが最急降下法と同じ欠点を持っていることが知られている。Rosenbrock<sup>[5]</sup>はHookeとJeevesの方法を発展させた座標回転法を提唱し、さらにNelderとMead<sup>[6]</sup>はSimplex法を提唱しているが、それらの計算効率がその中で使われる定数の定め方に左右されるにも拘わらずそれらの定め方に基準がないという欠点をもっている。

我々がここで提案する方法は考え方としてはSimplex法に類似したもので、Simplex法が変数が作る空間でのSimplexの端点における目的関数の値を評価し、Simplexの折り返し、縮小、拡大という基本操作によりこれを最適点へと動かして行くのに対し、我々の方法はその空間における超球上での関数値を評価しこれを最適点へと動かして行くものである。

なお、この研究はシステム解析手法研究グループの活動の一環としてなされたもので、この新しい手法は一つのサブルーチンとして最適化コードシステムSCOOPの中に組みこまれた。

## 2. アルゴリズムの概要

ここで述べる解法は、与えられた球面上にできるだけ一様になるように探索点を作り、それらの点における目的関数の値を評価し、最小値（以後最適化問題とは目的関数の最小値を求める問題と考えることにする）を与える方向へ球を移動させながら最適点を探索していく繰り返し法で、球面探索法と呼ぶことにする。

アルゴリズムの説明の前に以下の量が既知であると仮定する。

$F(x)$  : 変数  $x \in E^n$  の関数 (目的関数)。

$x_0$  : 変数  $x$  の初期値。球の最初の中心として  $x_0$  がとられる。

$r_0$  : 最初の球の半径。

$N$  : 一球面上の探索点の数。

$\alpha$  : 球の半径縮小係数。 ( $0 < \alpha < 1$ )

$\beta$  : 球を移動させるときの距離。 ( $\beta > 1$ )



2.1 探索点とその空間座標

原点を中心とした単位球面上の点（基点と呼ぶ）を  $P_0$  とすると、任意の位置にある、任意の半径の球面上の  $P_0$  に対応する点の座標は  $P_0$  の座標の一次変換によって求めることができる。それ故繰り返し計算の過程で用いる探索点の座標は、計算の初めに基点に対して以下のように求めておく。

(i)  $n$  (変数の数) = 2 (円) の場合

$$\begin{aligned} x_i &= (x_{1i}, x_{2i}) \\ &= (\cos \theta_i, \sin \theta_i) \end{aligned}$$

但し  $\theta_i$  は原点を始点とするベクトル  $\overrightarrow{OP_0}$  と  $x_1$  軸の成す角で、例えば  $\theta$  方向に等分割することにより、

$$\theta_i = i \cdot \frac{2\pi}{N}, \quad i = 1 \dots N$$

として求める。

(ii)  $n = 3$  (球) の場合

$$\begin{aligned} x_{ij} &= (x_{1ij}, x_{2ij}, x_{3j}) \\ &= (\cos \theta_i \cos \varphi_j, \sin \theta_i \cos \varphi_j, \sin \varphi_j) \end{aligned}$$

但し  $\theta_i$  は  $\overrightarrow{OP_0}$  と  $x_1 \cdot x_3$  平面の成す角、 $\varphi_j$  は  $\overrightarrow{OP_0}$  と  $x_1 \cdot x_2$  平面の成す角で (Fig. 1 参照)、 $\theta$  方向、 $\varphi$  方向に等分割すると探索点が  $x_3$  - 極の周りに密になるため、次のようにする。

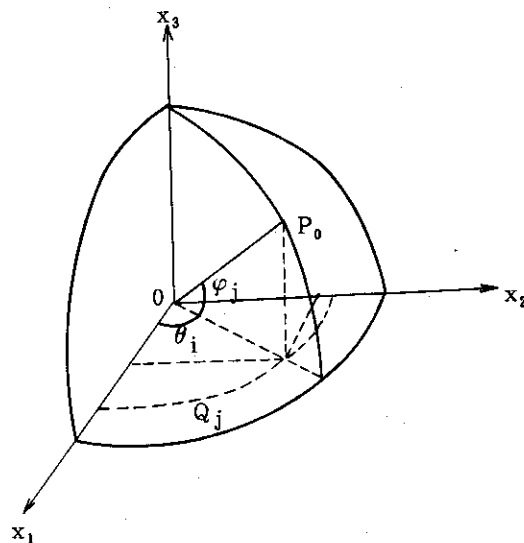


Fig. 1 Selection of basis points  $P_0$  on the surface of unit sphere

まず  $\varphi$  方向に対しては等分割する。

$$\varphi_j = j \cdot \frac{2\pi}{N_0}, \quad j=1, \dots, N_0$$

このとき  $x_1 \cdot x_2$  平面と  $\varphi_j$  の角を成す球上の点の  $x_1 \cdot x_2$  平面への像は  $r_j = \cos \varphi_j$  を半径とする円  $Q_j$  になる。そこで全ての  $Q_j$  の周の長さの和と  $Q_j$  自身の長さの比を重みとして  $Q_j$  上に  $N_j$  個の点を以下のように作る。

$$\left. \begin{aligned} W_j &= \cos \varphi_j / \sum_{j=1}^{N_0} \cos \varphi_j \\ N_j &= N \times W_j \\ Q_j &= i \cdot \frac{\pi}{N_j}, \quad i=1, \dots, N_j \end{aligned} \right\} \quad j=1, \dots, N_0$$

(iii)  $n=4$  (超球) の場合

$$X_{ijk} = (X_{1ijk}, X_{2ijk}, X_{3jk}, X_{4k})$$

但し

$$\begin{aligned} X_{1ijk} &= \cos \theta_i \cdot \cos \varphi_j \cdot \cos \phi_k, \\ X_{2ijk} &= \sin \theta_i \cdot \cos \varphi_j \cdot \cos \phi_k, \\ X_{3jk} &= \sin \varphi_j \cdot \cos \phi_k, \\ X_{4k} &= \sin \phi_k \end{aligned}$$

で、 $\theta_i, \varphi_j, \phi_k$  はそれぞれ  $\overrightarrow{OP_0}$  と  $x_1 \cdot x_4$ ,  $\overrightarrow{OP_0}$  と  $x_1 \cdot x_3$ ,  $\overrightarrow{OP_0}$  と  $x_1 \cdot x_2$  平面との成す角である。この場合の各  $\theta_i, \varphi_j, \phi_k$  の決定も  $n=3$  の場合と同様に行なう。

## 2.2 探索のアルゴリズム

2.1 節での探索点の座標が計算されたものとする。球面探索法のアルゴリズムは以下のようになる。

- (i) 最初の球の中心  $x_c$  を  $x_c = x_0$  とし、半径を  $r = r_0$  とする。また  $x_0$  における目的関数の値  $F_0 = F(x_0)$  を求める。
- (ii) 2.1 で求めよれた  $N$  個の点  $x_b$  を、半径  $r$ 、中心  $x_c$  の球面に対応させる。即ち

$$x_i = r \cdot x_b + x_c, \quad i=1 \dots N$$

とし、

$$F_m = \min_i F(x_i)$$

を求め、そのときの点を  $x_m$  とする。

若し、 $\|x_m - x_0\| < \epsilon$  なら (iv) へ行く。但し  $\epsilon$  はあらかじめ定めた許容量で、 $\|\cdot\|$  はユークリッドの距離を示す。

- (iii) (ii)で求めた  $F_m$  が  $F_m < F_0$  なら  $F_0 := F_m$ ,  $x_0 := x_m$  とし,  $x_m - x_c$  方向に球を移動させる。即ち新しい球の中心を次のように求める。

$$x_c = x_c + \beta \cdot (x_m - x_c)$$

但し  $\beta$  は移動幅で,  $A := B$  は A を B と置き換えることを意味する。Fig. 2 には  $\beta = 2$  の場合が示されている。

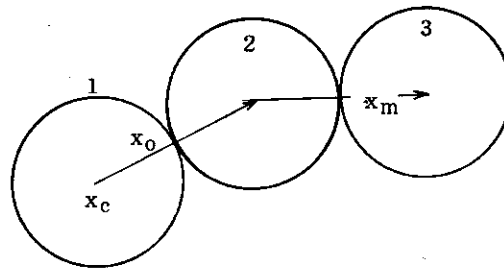


Fig. 2 Movement of searching sphere for the minimum objective when  $F(x_m) < F(x_0)$

- (iv)  $F_m \geq F_0$  なら  $x_c := x_0$  とし球の半径を

$$r = r e^{-\alpha}$$

と縮小する。この理由は最適点がこの段階での  $x_0$  の近傍に存在する可能性があり, 従って関数の等高線が狭くなっているだろうと予想されるからである (Fig. 3 参照)。

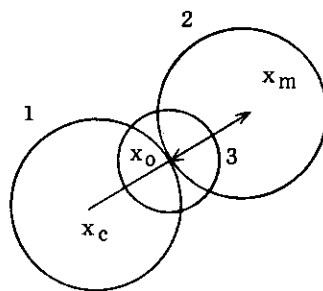


Fig. 3 Movement of searching sphere for the minimum objective when  $F(x_m) \geq F(x_0)$

- (iii), (iv)いずれの場合も(ii)へ戻って探索をくり返す。  
 (v) (ii)~(iv)までのくり返しの過程で, 球の半径があらかじめ定めた基準以下になったら計算を終了する。  
 以上のアルゴリズムのフローチャートを書くと Fig. 4 のようになる。

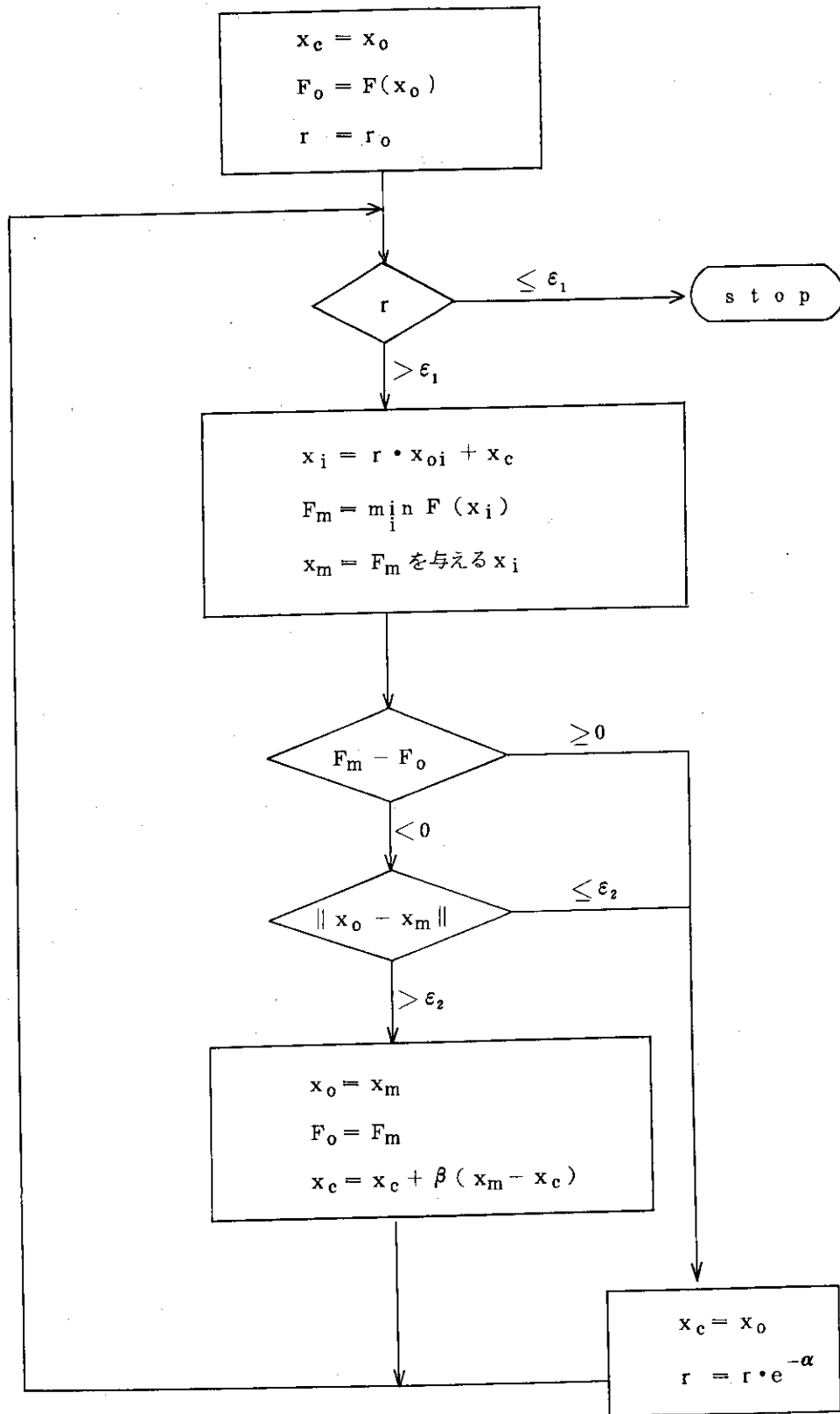


Fig. 4 Flow chart of algorithms of the spherical search method

Fig. 5 は二変数の問題に対する球面探索法の探索過程を Rosenbrock の関数,  $F(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ , を例に図示したもので破線がその等高線である。図のようにこの関数は  $x_2 = x_1^2$  に沿って狭く曲がった等高線を持ち,  $x_1 = x_2 = 1$  で最小値  $F = 0$  を持つことが知られている。初期設定として中心を  $x_0 = (0, 0)$  とし, 球の半径  $r_0 = 0.1$ , 半径縮小係数  $\alpha = 1.0$ , 移動幅  $\beta = 2.0$  として計算した結果の図である。以下この図について説明を行う。

- (1) まず原点を中心に半径 0.1 の円 1 の円周上での  $F$  の最小値を与える点  $x_0$  を探がす (2 の円との接点が求められた点)。次に原点と  $x_0$  を結ぶ方向に  $\beta = 2.0$  の距離に次の円の中心  $O_2$  を求めた。
  - (2) (1) で求めた点  $O_2$  を中心に同じ半径の円 2 が描かれ, この円周上での  $F$  の最小値を与える点 (3 の円との接点) が求められた。この時の  $F$  の値と円 1 上の最小値が比較され, 円 2 における値の方が小さかったので, 円 2 の中心  $O_2$  と最小値を与える点を結ぶ方向に次の円の中心  $O_3$  が求められた。
  - (3) 以下円 3 から円 8 までは(2)と同様の過程で描かれたもので, 図のように円が  $x_2 = x_1^2$  に沿って移動している。
  - (4) 円 8 での  $F$  の最小値を与える点が円 7 における点 (即ち円 7 と円 8 の接点) と非常に近い距離にあったため次の円 9 の中心は円 7 と円 8 の接点にとられ, 半径  $r = r e^{-\alpha}$  と縮小された。さらに円 10, 円 11 が(2)と同様の過程で描かれた。
- 以後結局 36 回の繰り返しで最適点に収束している。

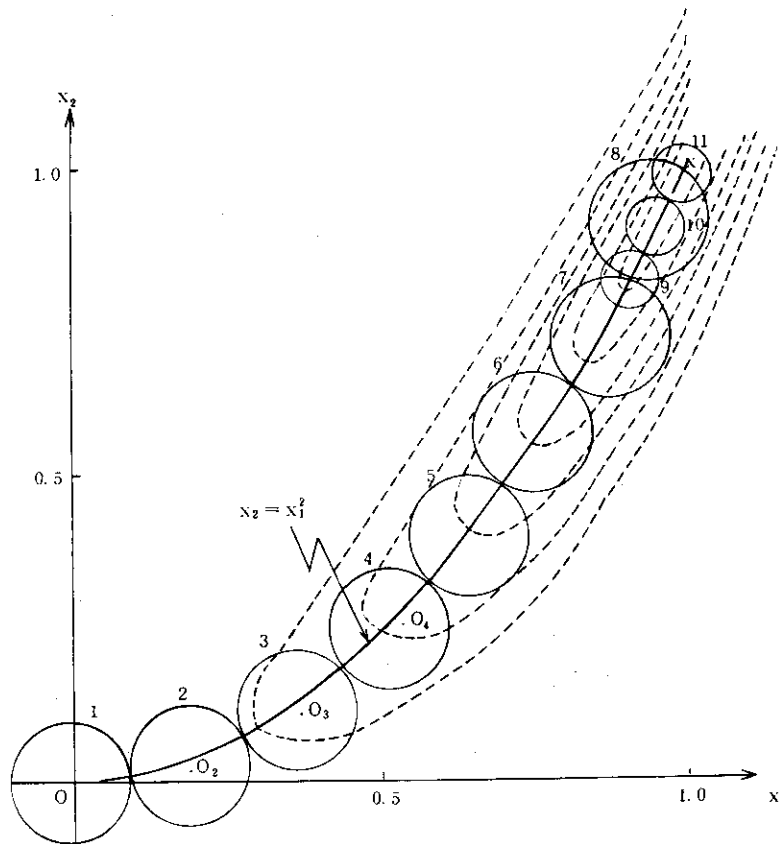


Fig. 5 Trajectory of searching sphere for finding the minimum of the Rosenbrock's function

## 3. テスト問題と計算結果

## 3.1 テスト問題

(i) Rosenbrock の関数<sup>[7]</sup>

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

この関数は Fig. 5 ですでに示したように  $x_2 = x_1^2$  にそって険しい谷があり,  $x = (1, 1)$  で最小値  $f = 0$  を持つ。

(ii) Beale の関数<sup>[8]</sup>

$$f(x) = \sum_{i=1}^3 (c_i - x_i(1 - x_2^i))^2$$

但し  $c_1 = 1.5$ ,  $c_2 = 2.25$ ,  $c_3 = 2.625$  である。この関数は  $x_2 = 1$  の近くに狭く曲がった谷があり,  $x = (3, 0.5)$  で最小値  $f = 0$  を持つ。

(iii) Box の関数<sup>[9]</sup>

$$f(x) = \sum_i \{ (e^{-x_1 y_i} - e^{-x_2 y_i}) - x_3 (e^{-y_i} - e^{-10 y_i}) \}^2$$

但し  $y_i = i/10$ ,  $i = 1 \dots 10$  である。  $f$  は  $x = (1, 10, 1)$ ,  $x = (10, 1, -1)$ ,  $(x_1 = x_2, x_3 \rightarrow 0)$ ,  $(x_1 \rightarrow +\infty, x_2 \rightarrow +\infty, x_3 \rightarrow 0)$  で最小値 0 をとる。

(iv) Gauss の関数<sup>[9]</sup>

$$f(x) = \sum_i (x_1 e^{-(z_i - x_3)^2 \cdot x_2 / 2} - y_i)^2$$

$$z_i = 3.5 - 0.5 \times (i - 1), \quad i = 1, \dots, 15$$

$$y_1 = y_{15} = 0.0009, \quad y_5 = y_{11} = 0.1295$$

$$y_2 = y_{14} = 0.0044, \quad y_6 = y_{10} = 0.2420$$

$$y_3 = y_{13} = 0.0175, \quad y_7 = y_9 = 0.3521$$

$$y_4 = y_{12} = 0.0540, \quad y_8 = 0.3989$$

この問題は点  $z_i$  において  $y_i$  なる値をとる曲線が正規分布をしているものとしてその平均と分散を求める問題である。  $f$  は  $x = (0.39896, 1.0, 0.0)$  のとき残差  $f = 1.12793277E-8$  をとる。

(v) エンチームの関数<sup>[10]</sup>

$$f(x) = \sum_{i=1}^n \left( v_i - \frac{x_1(y_i^2 + x_2 y_i)}{y_i^2 + x_3 y_i + x_4} \right)^2$$

但し  $v_i, y_i$  は次の値をとる。

i	1	2	3	4	5	6	7	8	9	10	11
$v_i$	0.1957	0.1947	0.1735	0.16	0.0844	0.0627	0.0456	0.0342	0.0323	0.0235	0.0246
$y_i$	4.0	2.0	1.0	0.5	0.25	0.167	0.125	0.1	0.0823	0.0714	0.0625

$f$  は  $x = (0.1928, 0.1916, 0.1234, 0.1362)$  で最小値  $f = 3.075 \times 10^{-4}$  をとる。

### 3.2 計算の条件

出発点は

- (i), (ii) に対しては  $x_1, x_2 \in [-100, 100]$
- (iii) に対しては  $x_1 \in [-10, 10], x_2 \in [0, 20], x_3 \in [-100, 100]$
- (iv) に対しては  $x_1 \in [-0.6, 1.4], x_2 \in [0, 2], x_3 \in [-1, 1]$
- (v) に対しては  $x_1, x_2, x_3, x_4 \in [-1, 1]$

の範囲内で乱数によりそれぞれ10点ずつ発生させた。なおこれらの問題を解くにあたって定めた収束判定量, 係数  $\alpha, \beta$ , 球の半径などはTable1にまとめた。以下にそれらの量の定め方の基準をあげる。

- (1) 探索点の数  $N$ ; これに対しては基準はなく, 多くするほど安定性を増すが反面計算時間が増加する。
- (2) 半径  $r_0$ ; 目的関数の変数の取り得る変域が広い場合には大きくする。一般に最適化問題では, 目的関数の変数が持つ物理的あるいは経済的な意味からその変域が既知な場合が多いが, 既知でない場合にはある程度大きめにとった方が無難であろう。Table1の値は  $r_0$  を決める上での一つの基準を与えている。
- (3) 半径縮小係数  $\alpha$ ; 目的関数の変動が変数の微小変化に敏感な場合は小さくとる。
- (4) 移動幅  $\beta$ ; 現在までのテスト計算ではほぼ2.0で良いと思われる。
- (5) 収束判定量  $\epsilon_1, \epsilon_2$ ; これらの量は求める解の精度との兼ねあいで定められるべきである。本報での球面探索法の計算効率を調べるために用いた他の手法はALPACK<sup>(11)</sup>の中に用意されているSimplex法, Pattern search法, 加速gradient法, Parallel tangent法とCojugate gradient法である。

### 3.3 計算結果

#### (i) Rosenbrock の問題

出発点

	1	2	3	4	5	6	7	8	9	10
$x_1$	63.67	-86.03	13.53	27.85	48.64	-4.22	74.57	78.88	23.25	-8.31
$x_2$	33.37	20.63	-18.14	16.69	-21.70	79.53	64.41	-82.36	-39.07	0.49

この10個の出発点に対応したRunNoに対して、本報の球面探索法を含めて6手法により計算した結果をTable2にまとめた。このTableでITNは収束に要した繰り返しの回数、IFNは関数の評価回数、F-finalは最終的な関数値、 $\times$ -finalは収束した点、Tは要した計算時間である。なおRunNoの左側に $\times$ の印のある場合には、そのRunは最適点に収束しなかったことを示している。

## (ii) Bealeの問題

出発点はRosenbrockの問題と同じである。この際の結果はTable3にまとめてあるが、収束しなかった場合が相当多いのが目につく。

## (iii) Boxの問題

出発点

	1	2	3	4	5	6	7	8	9	10
X <sub>1</sub>	6.37	2.06	-3.06	-5.36	3.78	-8.34	9.31	-5.89	3.38	-4.99
X <sub>2</sub>	13.34	19.31	7.97	13.36	3.71	5.58	4.39	1.64	3.61	5.25
X <sub>3</sub>	27.17	2.77	-46.24	84.00	82.50	85.57	25.57	28.52	-87.45	-35.13

この10個の出発点による計算結果がTable4にまとめられている。

## (iv) Gaussの問題

出発点

	1	2	3	4	5	6	7	8	9	10
X <sub>1</sub>	1.037	0.606	0.094	-0.136	0.778	-0.434	1.331	-0.189	0.738	-0.099
X <sub>2</sub>	1.33	1.981	0.797	1.336	0.371	0.558	0.439	0.164	0.361	0.525
X <sub>3</sub>	0.272	0.028	-0.462	0.84	0.825	0.856	0.256	0.285	-0.874	-0.351

計算結果はTable5にまとめた。すべての場合について収束している。

## (v) Enzymeの問題

出発点

	1	2	3	4	5	6	7	8	9	10
X <sub>1</sub>	0.64	0.98	-0.64	0.011	0.22	-0.29	0.32	-0.20	0.39	0.42
X <sub>2</sub>	0.33	0.028	-0.95	0.50	-0.13	-0.25	-0.32	0.90	0.14	0.38
X <sub>3</sub>	0.27	-0.66	0.45	0.93	-0.81	-0.93	-0.83	-0.80	-0.67	0.54
X <sub>4</sub>	0.63	-0.21	-0.73	-0.96	0.35	0.69	-0.13	-0.95	0.72	-0.17

これに対する結果はTable6にまとめられている。

以上の計算結果からわかるように球面探索法は全ての問題に対して最適点に収束したが、他の手法では特にBealeの問題とEnzymeの問題に対して安定性が悪かった。Bealeの問題では $x_1 \cdot x_2$ 平面の第四象限を出発点とするRunNo.2, 6, 10に対してはほとんど収束しておらず、その全てがこの象限にある $x_2=1$ に沿った狭い谷におちこんでしまっている。Fig. 6はRunNo.2の場合の探索の過程を加速gradient法と球面探索法について図示したもので、○印が加速



gradient法で×印が球面探索法である。また印に付いている数字はその点に達した時のくり返し回数である。図のように加速gradient法では20回目のくり返しで谷にはいりこみ以後どんどんこの谷を降下している。これに対し球面探索法では3回目から $x_1$ 軸に平行に正の方向に進み20回目ではほぼ最適点の近傍に達している。この2つの過程の相異が2つの手法の安定性の相異を顕著に表わしている。

### 3.4 収束効率の比較と検討

収束効率 (C. E.) を次の式で計算し各手法と比較した。

$$C. E. = \ell n \frac{|F_0 - F_m|}{|FF - F_m|} / T$$

但し  $F_0$  : 出発点における目的関数の値

$FF$  : 収束した点における目的関数の値

$F_m$  : 既知な目的関数の最小値

$T$  : 収束に要した計算時間

Table 2 から 6 までに示した 5 つのテスト問題に対する計算についての収束効率を Table 7 から 11 までにまとめた。但し比較がし易いように各 Run No. 毎にその中で最も効率が良かった C. E. の値を 1 と規格化してある。

この結果からわかるように、球面探索法は Rosenbrock と Gauss の関数に対しては他の手法と比較しても収束効率が良いが、Beale, Box と Enzyme の関数に対しては、例えばパターンサーチと比べて 1/10 程度の場合もある。結局球面探索法は非常に安定した手法ではあるが収束効率にまだ問題がある。Fig. 7 はこの様子を Box の関数の Run No. 1 について Simplex 法, 加速 gradient 法, Parallel tangent 法を比較したもので、図のように加速 gradient 法では 20 回のくり返し点から急速に最適点へ収束しており、Parallel tangent 法では 40 回目までのくり返しでは最適点からかなり離れた点にあったがそれ以後は急速に最適点へと収束している。また Simplex 法も 60 回目まではゆるやかな下降線をたどり、それ以後急速に最適点へ近づいて行くのがわかる。これに対して球面探索法は全般的に緩やかな降下をたどりながら最適点への近づいており、この降下のパターンが全ての場合に対してとられている。それ故収束の方向へ向った点からの収束の加速法が今後の課題である。

Table 1 Parameters used for solving 5 test functions

Test function	N	$r_0$	$\alpha$	$\beta$	$\epsilon_1$	$\epsilon_2$
Rosenbrock	50	10.0	0.1	2.0	$10^{-8}$	$10^{-8}$
Beale	50	10.0	0.1	2.0	$10^{-8}$	$10^{-8}$
Box	100	1.0	0.5	2.0	$10^{-8}$	$10^{-8}$
Gauss	100	0.1	0.5	2.0	$10^{-8}$	$10^{-8}$
Enzyme	250	0.5	0.5	2.0	$10^{-8}$	$10^{-8}$

Table 2 Numerical results of 10 runs of 6 programs for Rosenbrock's function

Program	Run No.	ITN	IFN	F-final	×-final	T(sec)
Spherical search	1	962	48051	0.2245-7*	10001, 1.0003	1.021
	2	268	13351	0.1216-8	0.99997, 0.99994	0.284
	3	264	13151	0.8158-8	10001, 1.0002	0.280
	4	252	12551	0.2890-8	10001, 1.0001	0.266
	5	366	18251	0.4727-8	0.99999, 0.99998	0.387
	6	1009	50401	0.2167-8	0.99997, 0.99993	1.068
	7	268	13351	0.4980-8	10001, 1.0001	0.283
	8	258	12851	0.2509-8	0.99999, 0.99998	0.273
	9	269	13401	0.1855-7	0.99988, 0.99976	0.283
	10	263	13101	0.1705-8	0.99996, 0.99992	0.277
Simplex (ALPACK)	1	143	668	0.4962-16	1.0, 1.0	0.854
	2	118	475	0.9057-17	"	0.475
	3	69	336	0.2267-16	"	0.414
	4	125	517	0.2883-16	"	0.517
	5	121	505	0.2663-16	"	0.674
	6	153	697	0.4025-16	"	0.872
	7	144	568	0.1714-16	"	0.797
	8	90	500	0.3524-16	"	0.585
	9	96	505	0.6660-16	"	0.506
	10	42	254	0.525-16	"	0.296
Pattern search (ALPACK)	1	680	1784	0.3274-12	1.0, 1.0	2.450
	2	522	1371	0.5869-9	100002, 100004	1.878
	3	238	772	0.7372-9	0.99997, 0.99995	0.885
	4	221	706	0.3674-11	1.0, 1.0	0.812
	5	88	390	0.1828-9	100001, 100003	0.350
	6	545	1503	0.1455-12	1.0, 1.0	2.005
	7	701	1765	0.1462-10	1.0, 1.0	2.537
	8	218	705	0.4547-11	1.0, 1.0	0.831
	9	133	486	0.8062-10	0.99999, 0.99998	0.512
	10	204	702	0.3274-12	1.0, 1.0	0.748

Table 2 (Continued)

Program	Run No.	ITN	IFN	F-final	×-final	T(sec)
Accelerated gradient (ALPACK)	1	350	1015	0.7749-19	1.0, 1.0	1.870
	2	351	968	0.7303-15	"	1.783
	3	110	354	0.4477-18	"	0.610
	4	307	1073	0.3658-18	"	1.753
	5	130	489	0.1338-16	"	0.782
	6	472	1283	0.1906-13	"	2.215
	7	389	1134	0.2491-18	"	1.959
	8	141	605	0.1104-16	"	1.044
	9	100	317	0.4842-18	"	0.565
	10	272	924	0.1717-14	"	1.481
Parallel tangent ×** (ALPACK)	1	661	1229	0.1301-9	100001, 100002	2.906
	2	997	2000	0.1959-1	1.139, 1.299	4.277
	3	438	1050	0.2723-6	0.9995, 0.9989	2.111
	4	873	2000	0.2563-2	0.950, 0.902	3.941
	5	833	1968	0.3274-6	0.9994, 0.9988	3.893
	6	949	2000	1.3435	-2.634, 6.988	4.083
	7	576	1145	0.2289-6	1.0065, 1.0010	2.373
	8	717	1717	0.3505-6	0.9998, 0.9996	3.163
	9	623	1475	0.4378-7	0.9994, 0.9988	2.759
	10	529	1264	0.1100-6	0.9997, 0.9993	2.354
Conjugate gradient (ALPACK)	1	366	948	0.3249-16	1.0, 1.0	1.716
	2	433	1061	0.5590-15	"	1.865
	3	141	504	0.5366-15	"	0.733
	4	278	818	0.7431-21	"	1.369
	5	88	384	0.1520-19	"	0.532
	6	542	1292	0.4253-15	"	2.387
	7	461	1125	0.1620-19	"	2.099
	8	85	385	0.5794-18	"	0.557
	9	142	489	0.6616-16	"	0.809
	10	133	473	0.1314-16	"	0.719

\* Should be read as  $0.2245 \times 10^{-7}$ 

\*\* Run marked with × did not converge to the optimal point

Table 3 Numerical results of 10 runs of 6 programs for Beale's function

Program	Run No.	ITN	IFN	F-final	×-final	T(sec)	
Spherical search	1	327	16301	0.9806-9	3.0, 0.5	0.423	
	2	281	14001	0.1239-8	3.0, 0.5	0.365	
	3	251	12501	0.1108-8	"	0.328	
	4	287	14301	0.1157-8	"	0.374	
	5	266	13251	0.1262-8	"	0.346	
	6	326	16251	0.7749-9	"	0.425	
	7	283	14101	0.1183-8	"	0.368	
	8	306	15251	0.1259-8	"	0.397	
	9	284	14151	0.1076-8	"	0.372	
	10	283	14101	0.1243-8	"	0.370	
Simplex (ALPACK)	1	130	494	0.6599-17	3.0, 0.5	0.395	
	×	2	630	1999	0.4521	-0.228+5, 1.0	1.653
	3	70	321	0.1335-15	3.0, 0.5	0.244	
	4	100	538	0.1072-15	"	0.372	
	5	120	490	0.4994-16	"	0.381	
	×	6	620	1998	0.4522	-0.841+4, 1.0	1.740
	7	135	540	0.7226-16	3.0, 0.5	0.435	
	8	135	524	0.5401-16	"	0.399	
	9	85	376	0.5885-16	"	0.284	
Pattern search (ALPACK)	×	10	625	1999	0.4521	-0.149+5, 1.0	1.740
	×	1	745	1996	0.4527	-0.216+4, 1.0	0.973
	×	2	790	1999	0.4524	-0.425+4, 1.0	0.974
	3	55	514	0.5741-11	3.0, 0.5	0.156	
	4	65	500	0.0	"	0.152	
	5	60	478	0.4093-10	299999, 049999	0.146	
	6	65	522	0.2558-11	3.0, 0.5	0.163	
	7	80	600	0.5121-14	"	0.181	
	×	8	710	1999	0.4527	-0.233+4, 1.0	0.916
	9	100	603	0.0	3.0, 0.5	0.195	
×	10	735	1996	0.4528	-0.183+4, 1.0	0.943	

Table 3 (Continued)

Program	Run No.	ITN	IFN	F-final	⊗-final	T(sec)	
Accelerated gradient (ALPACK)	×	1	400	1209	0.1097-21	3.0, 0.5	1.168
	×	2	235	842	0.4592	-0.21+3, 1.005	0.738
	×	3	200	716	0.1984-20	3.0, 0.5	0.696
	×	4	305	911	0.1568-21	"	0.868
	×	5	490	1550	0.6492-22	"	1.428
	×	6	620	2000	0.4961	-0.348+3, 1.03	1.589
	×	7	470	1867	0.2099-21	3.0, 0.5	1.720
	×	8	405	1257	0.2333-21	"	1.119
	×	9	190	607	0.1889-21	"	0.608
	×	10	660	2000	0.5405	-0.168+2, 1.05	1.889
Parallel tangent (ALPACK)	×	1	1565	2000	0.2843	9.4, 0.88	6.001
	×	2	15	89	0.470	-84.6, 1.01	0.162
	×	3	240	397	0.8972-14	3.0, 0.5	1.068
	×	4	320	525	0.1390-14	"	1.428
	×	5	885	1184	0.2461-15	"	3.467
	×	6	10	82	7.162	-0.54-5, 79.01	0.146
	×	7	1250	1631	0.7608-14	3.0, 0.5	4.829
	×	8	885	1213	0.1435-13	"	3.432
	×	9	245	384	0.5519-17	"	1.017
	×	10	1355	1783	0.469	-89.5, 1.01	4.952
Conjugate gradient (ALPACK)	×	1	865	2000	0.47	-84.74, 1.01	1.566
	×	2	865	2000	0.464	-126.6, 1.008	1.507
	×	3	915	2000	7.338	0.26-7, -0.46+3	1.634
	×	4	895	2000	0.4614	-163.6, 1.006	1.516
	×	5	890	2000	7.338	0.25-7, -0.47+3	1.588
	×	6	920	2000	0.4944	-35.7, 1.027	1.551
	×	7	910	2000	0.4875	-43.47, 1.022	1.574
	×	8	895	2000	7.327	0.456-8, -0.83+4	1.645
	×	9	905	2000	7.333	0.13-7, -0.58+3	1.649
	×	10	65	343	0.2442-20	3.0, 0.5	0.299

Table 4 Numerical results of 10 suns of 6 programs for Box's function

Program	Run No.	ITN	IFN	F-final	⊗-final	T(sec)	
Spherical search	1	156	14881	0.4293-8	1.0002,9.997,0.9999	8.727	
	2	164	15649	0.5433-8	0.9998,10.003,1.0001	9.246	
	3	393	37633	0.6555-7	3.035, 3.026, -0.15-2	20.224	
	4	216	20641	0.5422-8	0.9998,10.003,1.0001	12.147	
	5	201	19201	0.1001-7	0.9997,10.005,1.0002	11.301	
	6	208	19873	0.8476-8	0.9997,10.004,1.0002	11.710	
	7	367	35137	0.5747-7	2.999, 3.007, 0.14-2	18.813	
	8	173	16513	0.4534-8	1.0002,9.997,0.9999	9.762	
	9	194	18529	0.5237-8	10.003,0.999,-1.0001	10.960	
	10	364	34849	0.8206-7	3.004, 2.995, -0.16-2	18.621	
Simplex (ALPACK)	1	165	1998	0.0	20.6, 20.6, 0.495-26	11.838	
	2	80	471	0.4751-17	1.0, 1.0, 1.0	2.752	
	3	345	1998	0.0	125.2, 125.2, 0.47-38	18.111	
	4	130	1998	0.2664-32	3.97, 3.97, 0.703-16	11.432	
	5	145	641	0.9270-17	1.0, 1.0, 1.0	3.787	
	6	245	1043	0.1199-16	"	6.173	
	7	150	1999	0.3126-36	15.38, 15.38, -0.53-18	11.648	
	8	165	913	0.3546-16	1.0, 1.0, 1.0	5.290	
	9	170	726	0.8321-17	1.0, 1.0, -1.0	4.373	
	10	360	1999	0.0	200.8, 200.8, 0.22-38	20.243	
Pattern search (ALPACK)	×	1					
		2	130	882	0.1333-32	1.0, 1.0, 1.0	4.607
	×	3	70	742	0.7559-1	0.441+3, 0.614, -1.32	8.148
		4	175	1093	0.1420-71	0.592+3, 0.592+3, 0.0	21.381
	×	5					
		6	210	984	0.4098-18	3.058, 3.058, 0.372-8	5.351
	×	7	60	650	0.7559-1	0.468+3, 0.614, -1.32	8.061
	×	8					
		9	120	1205	0.0	482.1, 482.1, 0.0	23.242
		10	130	1317	0.0	599.5, 599.5, 0.0	26.922

Table 4 (Continued)

Program	Run No.	ITN	IFN	F-final	⊗-final	T(sec)
Accelerated gradient (ALPACK)	1	110	610	0.1470-25	9.18, 9.18, -0.94-13	3.746
	2	435	1441	0.2081-20	1.0, 10.0, 1.0	9.014
	3	650	2000	0.1012-18	47.7, 47.7, -0.599-10	12.843
	4	595	1746	0.5425-15	1.0, 10.0, 1.0	11.055
	5	450	1171	0.1279-16	"	7.450
	6	400	1391	0.7711-16	"	8.722
	7	255	2000	0.3045-24	7.86, 7.86, 0.37-12	12.960
	8	545	1560	0.1566-19	1.0, 10.0, 1.0	9.550
	9	675	1999	0.3403-14	10.0, 1.0, -1.0	12.630
	10	410	1115	0.3399-16	"	6.920
Parallel tangent (ALPACK)	1	140	249	0.2138-16	10.75, 10.75, -0.21-8	2.288
	2	465	687	0.1043-12	1.0, 10.0, 1.0	6.424
	3	975	1998	0.1724-10		16.688
	4	585	833	0.4703-15	1.0, 10.0, 1.0	7.649
	5	870	1227	0.1712-13	"	11.054
	6	690	978	0.4392-13	"	8.910
	7	240	377	0.8024-15	7.65, 7.65, -0.24-7	3.468
	8	735	1040	0.2902-12	1.0, 10.0, 1.0	9.627
	9	835	1183	0.4141-14	10.0, 1.0, -1.0	10.600
	10	845	1199	0.8800-14	"	11.364
⊗ Conjugate ⊗ gradient ⊗ (ALPACK)	1	200	691	0.1375-26	61.77, 61.77, -0.98-14	4.052
	2					
	3	205	1153	0.5115-32	6.742, 6.742, -0.72-16	6.899
	4					
	5					
	6	100	868	0.7559-1	0.614, 0.541+4, 1.32	13.063
	7	160	666	0.3020-25	33.98, 33.98, 0.35-13	3.929
	8	500	2000	0.7559-1	0.613, 0.2+3, 1.32	17.304
	9	300	987	0.1648-19	10.0, 1.0, -1.0	5.805
	10	195	1297	0.8961-22	1.0, 10.0, 1.0	7.530

Table 5 Numerical results of 10 runs of 6 programs for Gaussian function

Program	Run No.	ITN	IFN	F-final	⊗-final	T(sec)
Spherical search	1	53	4993	0.1128-7	0.3989, 10, 0.164-4	1.507
	2	53	4993	"	" , 0.407-4	1.505
	3	51	4801	"	" , 0.311-4	1.447
	4	56	5281	"	" , 0.294-4	1.592
	5	54	5089	"	" , 0.388-4	1.536
	6	56	5281	"	" , 0.130-4	1.590
	7	58	5473	"	" , 0.241-4	1.638
	8	53	4993	"	" , 0.427-4	1.492
	9	52	4897	"	" , 0.373-4	1.462
	10	50	4705	"	" , -0.387-4	1.405
Simplex (ALPACK)	1	65	1998	0.1128-7	0.3989, 1.0, -0.101-10	6.632
	2	75	1999	0.1128-7	" " -0.597-11	6.536
	3	75	894	0.1128-7	" " 0.135-10	2.961
	4	75	1998	0.1128-7	" " -0.296-11	6.334
	5	70	901	0.1128-7	" " -0.161-10	2.877
	6	85	1998	0.1128-7	" " -0.421-11	6.330
	7	80	1998	0.1128-7	" " -0.584-11	6.448
	8	75	1998	0.1128-7	" " -0.422-10	6.190
	9	75	1999	0.1128-7	" " 0.357-10	6.514
	10	65	1998	0.1128-7	" " -0.222-10	6.426
Pattern search (ALPACK)	1	60	760	0.1128-7	0.3989, 10, 0.0	2.125
	2	45	694	"	"	1.861
	3	50	715	"	"	1.891
	4	50	717	"	"	1.861
	5	45	709	"	"	1.945
	6	60	786	"	"	2.041
	7	45	723	"	"	1.887
	8	50	689	"	" " -0.455-11	1.835
	9	45	687	"	" " 0.0	1.825
	10	45	682	"	"	1.840



Table 5 (Continued)

Program	Run No.	ITN	IFN	F-final	X-final	T(sec)
Accelerated gradient (ALPACK)	1	145	504	0.1128-7	0.3989, 10, -0.569-9	1.920
	2	185	541	"	" " , -0.613-8	1.992
	3	190	609	"	" " , 0.190-9	2.205
	4	185	541	"	" " , 0.148-8	1.952
	5	230	716	"	" " , 0.887-9	2.593
	6	230	712	"	" " , -0.360-8	2.599
	7	135	413	"	" " , -0.14 -8	1.481
	8	265	870	"	" " , -0.284-8	3.625
	9	185	573	"	" " , -0.577-8	2.142
	10	140	461	"	" " , -0.139-10	1.676
Paralled tangent (ALPACK)	1	120	247	0.1128-7	0.3989, 10, 0.799-8	1.464
	2	155	296	"	" " , -0.54 -8	1.780
	3	130	255	"	" " , 0.12 -8	1.485
	4	120	225	"	" " , 0.23 -7	1.325
	5	195	358	"	" " , -0.57 -10	2.138
	6	155	291	"	" " , 0.13 -7	1.719
	7	155	294	"	" " , 0.54 -9	1.703
	8	135	288	"	" " , 0.29 -9	1.635
	9	140	273	"	" " , -0.11 -7	1.563
	10	130	250	"	" " , 0.93 -8	1.395
Conjugate gradient (ALPACK)	1	45	310	0.1128-7	0.3989, 10, 0.172-11	1.144
	2	50	312	"	" " , 0.484-14	1.155
	3	50	319	"	" " , 0.505-14	1.162
	4	40	309	"	" " , 0.443-14	1.127
	5	60	363	"	" " , -0.127-11	1.325
	6	55	332	"	" " , -0.119-12	1.212
	7	70	376	"	" " , -0.498-11	1.345
	8	85	420	"	" " , -0.481-14	1.472
	9	65	355	"	" " , 0.870-13	1.262
	10	75	371	"	" " , -0.194-11	1.310

Table 6 Numerical results of 10 runs of 6 programs for Enzyme's function

Program	Run No.	ITN	IFN	F-final	×-final	T(sec)	
Spherical search	1	121	30001	0.3104-3	0.1928, 0.1932 0.1245, 0.1366	3.233	
	2	70	17251	0.3104-3	0.1927, 0.1944 0.1247, 0.1372	1.854	
	3	425	105341	0.3104-3	0.1927, 0.1947 0.1248, 0.1373	11.410	
	4	117	29001	0.3104-3	0.1928, 0.1939 0.1245, 0.1370	3.124	
	5	60	14751	0.3104-3	0.1927, 0.1948 0.1247, 0.1374	1.588	
	6	179	44501	0.3104-3	0.1927, 0.1945 0.1246, 0.1372	4.793	
	7	725	181001	0.3104-3	0.1927, 0.1956 0.1249, 0.1377	19.481	
	8	202	50251	0.3104-3	0.1926, 0.1968 0.1252, 0.1382	5.421	
	9	70	17251	0.3104-3	0.1929, 0.1910 0.1240, 0.1356	1.856	
	10	156	38751	0.3104-3	0.1927, 0.1944 0.1248, 0.1372	4.166	
Simplex (ALPACK)	1	130	815	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	1.527	
	2	130	846	0.3075-3	"	1.519	
	×	3	145	813	0.1594-2	0.2339, -1.292 -0.836, -0.550	1.458
	×	4	475	1999	0.3555-1	1.06+5, -1.0 1.6+6, -1.6+6	3.751
	×	5	70	677	0.3075-3	same as Run No.1	1.215
	×	6	90	762	0.3075-3	"	1.396
	×	7	100	743	0.3075-3	"	1.357
	×	8	170	932	0.1594-2	same as Run No.3	1.773
	×	9	125	839	0.3075-3	same as Run No.1	1.630
	×	10	90	682	0.3075-3	"	1.336
Pattern search (ALPACK)	1	325	1511	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	1.426	
	×	2	690	1997	0.9524-3	-0.021, -1.317 8.366, 5.601	2.083
	×	3	645	1977	0.4252-3	0.226, -0.418 -0.021, -0.182	2.074
	×	4	695	1993	0.3750-2	0.294, -0.555 0.496, -0.549	2.118
	×	5	350	1696	0.3075-3	same as Run No.1	1.621
	×	6	475	1994	0.3075-3	"	1.984
	×	7	465	1859	0.4242-3	0.225, -0.415 -0.024, -0.178	1.862
	×	8	515	1993	0.3075-3	same as Run No.1	1.914
	×	9	700	1997	0.8799-3	0.086, 1.485 3.85, 2.71	2.043
	×	10	390	1668	0.3075-3	same as Run No.1	1.545

Table 6 (Continued)

Program	Run No.	ITN	IFN	F-final	X-final	T(sec)	
Accelerated gradient (ALPACK)	1	530	2000	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	3.324	
	×	2	290	1301	0.7264-3	0.236, -0.545 -0.059, -0.253	1.884
	×	3	145	954	0.7723-3	0.240, -0.545 -0.013, -0.277	1.330
	×	4	145	831	0.1425-1	0.363, -0.975 0.910, -1.86	1.184
	×	5	810	2000	0.1028-2	0.128, 1.082 50, 262	3.669
		6	605	1996	0.3075-3	same as Run No.1	3.419
	×	7	410	2000	0.1742-2	0.230, -0.954 -0.551, -0.387	2.906
	×	8	505	1843	0.1084	0.682-2, 0.248 -1.135, -1.575	2.863
		9	635	2000	0.3075-3	same as Run No.1	3.319
	×	10	155	793	0.4795-2	0.288, -0.421 0.596, -0.472	1.097
Parallel tangent (ALPACK)	1	1130	1550	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	6.637	
		2	1515	1997	0.5896-3	0.1686, 1.553 0.7243, 0.6196	8.275
	×	3	725	1054	0.1594-2	0.234, -1.292 -0.836, -0.55	4.452
		4	1175	1610	0.4242-3	0.225, -0.414 -0.024, -0.178	6.666
		5	845	1165	0.3075-3	same as Run No.1	4.912
	×	6	1570	1998	0.1008-2	-0.214, -2.23 14.3, 983	8.648
		7	1510	1996	0.3460-3	0.189, 0.328 0.159, 0.203	8.344
	×	8	1260	1729	0.1594-2	0.234, -1.29 -0.836, -0.55	6.764
		9	1390	1899	0.3075-3	same as Run No.1	7.713
		10	1500	1998	0.6052-3	0.163, 1.71 0.724, 0.663	8.275
Conjugate gradient (ALPACK)	1	310	1101	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	1.685	
	×	2	325	1179	0.1594-2	0.234, -1.292 -0.836, -0.55	1.748
	×	3	355	2001	0.6023-3	0.237, -0.509 -0.023, -0.245	2.518
	×	4	585	2000	0.4471-3	0.223, -0.388 -0.023, -0.161	2.704
		5	195	814	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	1.159
		6	250	1032	"	"	1.488
	×	7	195	1810	0.2499-2	0.21, -0.99 -0.69, -0.29	2.164
	×	8	220	1692	0.1684-2	0.23, -1.03 -0.62, -0.42	2.180
		9	245	941	0.3075-3	0.1928, 0.1909 0.1231, 0.1358	1.376
	×	10	480	1993	0.4242-3	0.22, -0.41 -0.024, -0.177	2.757

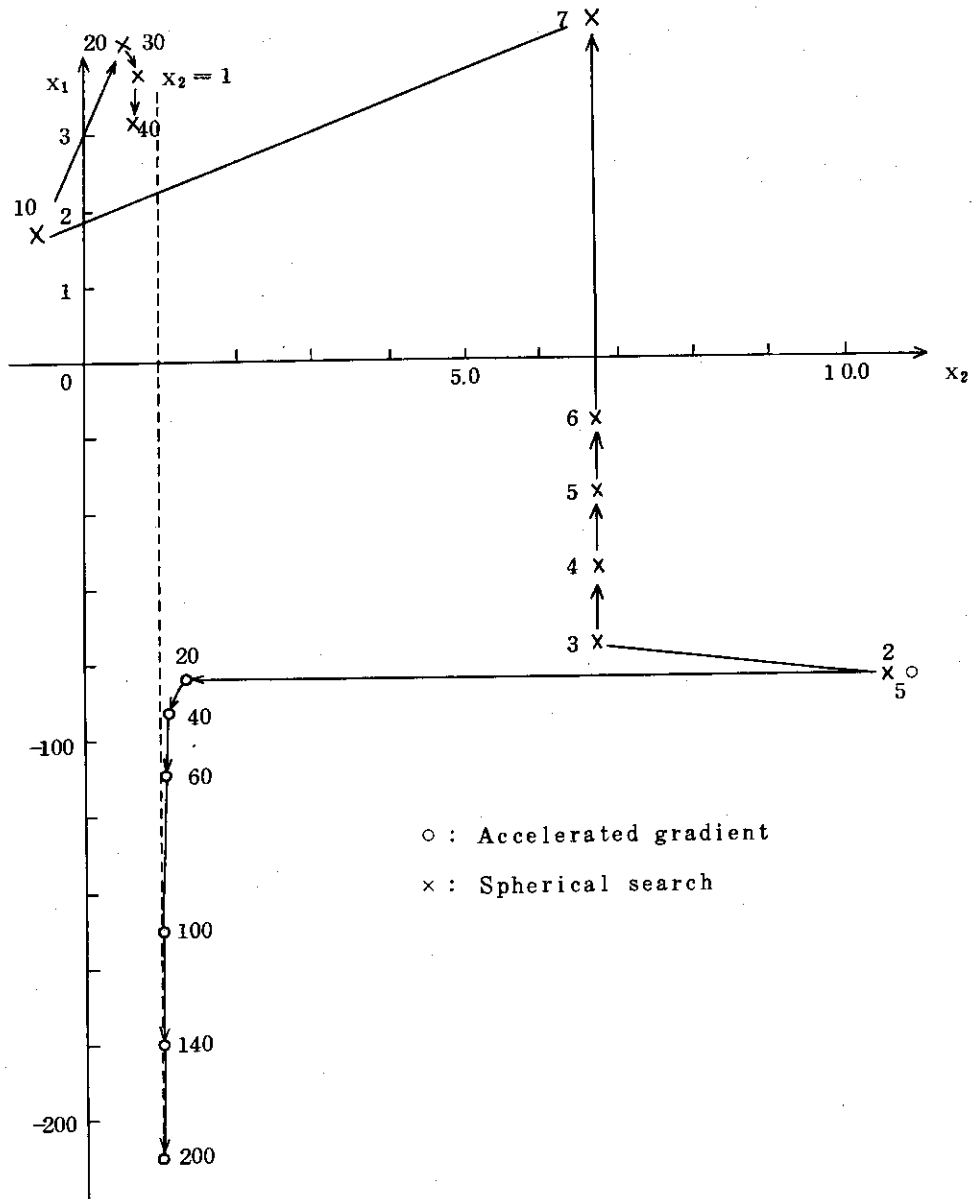


Fig. 6 Trajectories of searching process of Run No.2 for Beale's function by the accelerated gradient and the spherical search methods

Table 7 Normalized computation efficiencies of 10 runs of 6 programs for Rosenbrock's function

Run No. Program	1	2	3	4	5	6	7	8	9	10
Sph. Se.	0.552	1.0	0.934	1.0	0.822	0.529	1.0	1.0	1.0	0.693
Simplex	1.0	0.858	1.0	0.766	0.699	1.0	0.524	0.642	0.868	1.0
Pat. Se.	0.295	0.153	0.316	0.385	0.983	0.363	0.127	0.377	0.639	0.322
Acc. Gr.	5.05	0.202	0.728	0.243	0.610	0.344	0.228	0.381	0.849	0.182
Par. Tan.	0.219	0.041	0.110	0.042	0.072	—	0.107	0.075	0.099	0.070
C. Gr.	0.481	0.179	0.531	0.344	1.0	0.394	0.221	0.749	0.543	0.416

Table 8 Normalized computation efficiencies of 10 runs of 6 programs for Beale's function

Run No. Program	1	2	3	4	5	6	7	8	9	10
Sph. Se.	0.679	1.0	0.423	0.091	0.393	0.344	0.400	0.770	0.133	0.400
Simplex	1.0	—	0.779	0.130	0.487	—	0.442	1.0	0.227	—
Pat. Se.	—	—	1.0	1.0	1.0	1.0	1.0	—	1.0	—
Acc. Gr.	0.392	—	0.324	0.065	0.157	—	0.131	0.416	0.124	—
Par. Tan.	—	—	0.165	0.031	0.052	—	0.037	0.107	0.064	—
C. Gr.	—	—	—	—	—	—	—	—	—	1.0

Table 9 Normalized computation efficiencies of 10 runs of 6 programs for Box's function

Run No. Program	1	2	3	4	5	6	7	8	9	10
Sph. Se.	0.154	0.136	0.099	0.292	0.194	0.236	0.077	0.340	0.232	0.155
Simplex	0.756	0.911	0.826	0.914	1.0	0.818	0.464	1.0	1.0	1.0
Pat. Se.	—	1.0	—	1.0	—	1.0	—	—	0.696	0.755
Acc. Gr.	0.890	0.325	0.338	0.475	0.503	0.554	0.291	0.638	0.303	0.777
Par. Tan.	1.0	0.295	0.165	0.731	0.286	0.481	0.726	0.446	0.357	0.411
C. Gr.	0.850	—	1.0	—	—	—	1.0	—	0.848	0.899

Table 10 Normalized computation efficiencies of 10 runs of 6 programs for Gaussian function

Run No. Program	1	2	3	4	5	6	7	8	9	10
Sph. Se.	0.758	0.770	0.802	0.709	0.862	0.762	0.821	0.986	0.863	0.932
Simplex	0.172	0.176	0.392	0.177	0.460	0.191	0.208	0.236	0.193	0.203
Pat. Se	0.540	0.620	0.614	0.605	0.680	0.594	0.713	0.802	0.691	0.711
Acc. Gr.	0.595	0.579	0.526	0.577	0.511	0.466	0.908	0.405	0.589	0.781
Par. Tan.	0.781	0.648	0.781	0.850	0.619	0.705	0.790	0.900	0.804	0.938
C. Gr.	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 11 Normalized computation efficiencies of 10 runs of 6 programs for Enzyme's function

Run No. Program	1	2	3	4	5	6	7	8	9	10
Sph. Se.	0.239	0.511	1.0	1.0	0.360	0.159	0.041	0.196	0.391	0.191
Simplex	0.935	1.0	—	—	0.954	1.0	1.0	—	0.846	1.0
Pat. Se.	1.0	—	—	—	0.714	0.705	—	1.0	—	0.861
Acc. Gr.	0.425	—	—	—	—	0.411	—	—	0.410	—
Par. Tan.	0.213	0.179	—	0.885	0.234	—	0.163	—	0.179	0.159
C. Gr.	0.845	—	—	—	1.0	0.938	—	—	1.0	—

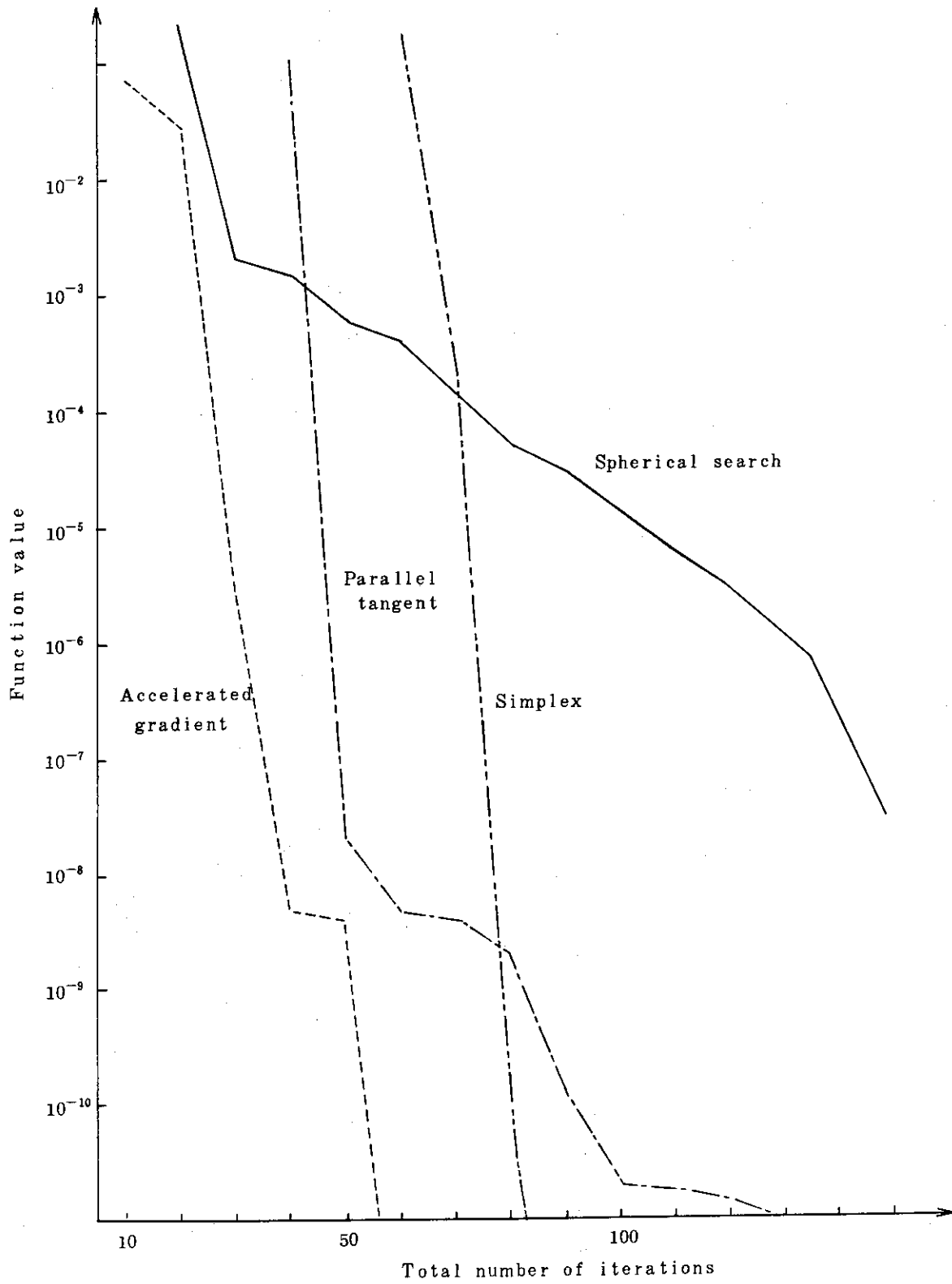


Fig. 7 Change of function value at each iteration process of Run No.1 for Box's function by four different methods

## 4. 結 論

最適化問題を解く上で大切なのは、用いる手法が出発点の位置に関係なく大域的な最適点を探索してくれること、即ち安定した手法が与えられることと、収束効率が良いことであろう。目的関数が単峰性でその等高線が緩やかな場合にはどの手法でも比較的安定していて、収束効率が良いが第三節で示したように、Bealeの関数やEnzymeの関数のようにその等高線がある直線に沿って深い谷を持ったり、狭く曲がっている場合、多くの手法がその収束性を出発点の位置に依存してしまう。この理由はそれらの手法がその探索方向をあらかじめ与えられた、あるいは反復の過程で計算された1つの方向に沿って行なうため、一たん谷の中には入りこむとどんどんとその谷を降下して行きなかなか脱出できないことにある。これに対し球面探索法では与えられた点を包む十分密な方向に対して探索を行ない、球の移動が移動幅 $\beta$ によって行われるため一たん谷の中には入りこんでも容易にそこから脱出できるため上述のような問題がおきず、また計算結果が示すように非常に安定した解法である。反面収束効率の点では最急降下法や共役傾斜法などの傾斜法に比較して優れているとは言えず、また他の直接探索法がそうであるように最適点の近傍での収束が遅いという欠点がある。以上のことから最適問題の解法として球面探索法を用いる際には、計算の初期の段階での大域的な粗い最小点の探索に用い、その点から局所的な最適点の探索には効率の良い手法を用いることが望ましい。

## 謝 辞

本手法の開発にあたり、終始貴重な御助言を下された原子炉工学部・篠原慶邦主任研究員に深く感謝いたします。

## 参 考 文 献

- (1) Fletcher R. and Reeves C. M.: "Function Minimization by Conjugate Gradients", Computer J., 7 (1964)
- (2) Davidon W. C.: "Variable Metric Method for Minimization", ANL-5990 (1959)
- (3) Fletcher R. and Powell M. J. D.: "A Rapidly Convergent Descent Method for Minimization", Computer J., 6 (1963)
- (4) Hooke R. and Jeeves T. A.: "Direct Search Solution of Numerical and Statistical Problems", J. Assoc. Comput. Math., 8 (1961)
- (5) Rosenbrock H. H.: "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3 (1960)



## 4. 結 論

最適化問題を解く上で大切なのは、用いる手法が出発点の位置に関係なく大域的な最適点を探索してくれること、即ち安定した手法が与えられることと、収束効率が良いことであろう。目的関数が単峰性でその等高線が緩やかな場合にはどの手法でも比較的安定していて、収束効率が良いが第三節で示したように、Bealeの関数やEnzymeの関数のようにその等高線がある直線に沿って深い谷を持ったり、狭く曲がっている場合、多くの手法がその収束性を出発点の位置に依存してしまう。この理由はそれらの手法がその探索方向をあらかじめ与えられた、あるいは反復の過程で計算された1つの方向に沿って行なうため、一たん谷の中にはいりこむとどんどんとその谷を降下して行きなかなか脱出できないことにある。これに対し球面探索法では与えられた点を包む十分密な方向に対して探索を行ない、球の移動が移動幅 $\beta$ によって行われるため一たん谷の中にはいりこんでも容易にそこから脱出できるため上述のような問題がおきず、また計算結果が示すように非常に安定した解法である。反面収束効率の点では最急降下法や共役傾斜法などの傾斜法に比較して優れているとは言えず、また他の直接探索法がそうであるように最適点の近傍での収束が遅いという欠点がある。以上のことから最適問題の解法として球面探索法を用いる際には、計算の初期の段階での大域的な粗い最小点の探索に用い、その点から局所的な最適点の探索には効率の良い手法を用いることが望ましい。

## 謝 辞

本手法の開発にあたり、終始貴重な御助言を下された原子炉工学部・篠原慶邦主任研究員に深く感謝いたします。

## 参 考 文 献

- (1) Fletcher R. and Reeves C. M.: "Function Minimization by Conjugate Gradients", Computer J., 7 (1964)
- (2) Davidon W. C.: "Variable Metric Method for Minimization", ANL-5990 (1959)
- (3) Fletcher R. and Powell M. J. D.: "A Rapidly Convergent Descent Method for Minimization", Computer J., 6 (1963)
- (4) Hooke R. and Jeeves T. A.: "Direct Search Solution of Numerical and Statistical Problems", J. Assoc. Comput. Math., 8 (1961)
- (5) Rosenbrock H. H.: "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3 (1960)

## 4. 結 論

最適化問題を解く上で大切なのは、用いる手法が出発点の位置に関係なく大域的な最適点を探索してくれること、即ち安定した手法が与えられることと、収束効率が良いことであろう。目的関数が単峰性でその等高線が緩やかな場合にはどの手法でも比較的安定していて、収束効率が良いが第三節で示したように、Bealeの関数やEnzymeの関数のようにその等高線がある直線に沿って深い谷を持ったり、狭く曲がっている場合、多くの手法がその収束性を出発点の位置に依存してしまう。この理由はそれらの手法がその探索方向をあらかじめ与えられた、あるいは反復の過程で計算された1つの方向に沿って行なうため、一たん谷の中には入りこむとどんどんとその谷を降下して行きなかなか脱出できないことにある。これに対し球面探索法では与えられた点を包む十分密な方向に対して探索を行ない、球の移動が移動幅 $\beta$ によって行われるため一たん谷の中には入りこんでも容易にそこから脱出できるため上述のような問題がおきず、また計算結果が示すように非常に安定した解法である。反面収束効率の点では最急降下法や共役傾斜法などの傾斜法に比較して優れているとは言えず、また他の直接探索法がそうであるように最適点の近傍での収束が遅いという欠点がある。以上のことから最適問題の解法として球面探索法を用いる際には、計算の初期の段階での大域的な粗い最小点の探索に用い、その点から局所的な最適点の探索には効率の良い手法を用いることが望ましい。

## 謝 辞

本手法の開発にあたり、終始貴重な御助言を下された原子炉工学部・篠原慶邦主任研究員に深く感謝いたします。

## 参 考 文 献

- (1) Fletcher R. and Reeves C. M.: "Function Minimization by Conjugate Gradients", Computer J., 7 (1964)
- (2) Davidon W. C.: "Variable Metric Method for Minimization", ANL-5990 (1959)
- (3) Fletcher R. and Powell M. J. D.: "A Rapidly Convergent Descent Method for Minimization", Computer J., 6 (1963)
- (4) Hooke R. and Jeeves T. A.: "Direct Search Solution of Numerical and Statistical Problems", J. Assoc. Comput. Math., 8 (1961)
- (5) Rosenbrock H. H.: "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3 (1960)

- (6) Nelder J. A. and Mead R.: "A Simplex Method for Function Minimization",  
Computer J., 7 (1965)
- (7) Dixon L. C. W.: "Nonlinear Optimization", CRANE, RUSSAK&COMPANY, INC.  
(1972)
- (8) コワリック, J., オスボーン, M. R.: "非線形最適化問題", 培風館 (1968)
- (9) Hillstrom K. E.: "A Simulation Test Approach to the Evaluation and  
Comparison of Unconstrained Nonlinear Optimization Algorithms",  
ANL-76-20 (1976)
- (10) Davidon W. C.: "Variance Algorithm for Minimization", Computer J., 10  
(1968)
- (11) Fromm N. C.: "ALRACK Reference Manual", Electronic Associates, INC.,  
Princeton, New Jersey. (1970)