

JAERI-M

7 2 5 5

座標回転法による非線形最適化プログラム
R O T A X

1 9 7 7 年 9 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.

座標回転法による非線形最適化プログラム

ROTAX

日本原子力研究所東海研究所原子炉工学部

鈴木 忠和

(1977年8月11日受理)

座標回転法を用いて制約条件を持つ非線形最適化問題を解くプログラムを開発し、手法の安定性、収束効率に対する評価を行った。座標回転法の特徴は直交座標系を常に目的関数が小さく（あるいは大きく）なる方向に回転させながら最適点を探索して行く直接探索法で、探索方向が変数を作る多次元空間において自由に回転することから目的関数の等高線が深い曲がった谷を持つ問題に対して効果的である。

制約条件を持つ問題へこの手法を適用するためにR.R.AllranとS.E.J. Johnsenにより提唱された方法を改善して用いた。その方法は制約条件式をもとの目的関数の中にペナルティ項として付加した新しい目的関数を考え、その目的関数を制約条件無しの最適化問題として解くことに帰着させるものである。

新しく開発されたプログラム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。

JAERI-M 7255

ROTAX: A Nonlinear Optimization Program by Axes Rotation Method

Tadakazu SUZUKI

Division of Reactor Engineering, Tokai Research

Establishment, JAERI

(Received August 11, 1977)

A nonlinear optimization program employing the axes rotation method has been developed for solving nonlinear problems subject to nonlinear inequality constraints and its stability and convergence efficiency were examined.

The axes rotation method is a direct search of the optimum point by rotating the orthogonal coordinate system in a direction giving the minimum objective. The searching direction is rotated freely in multi-dimensional space, so the method is effective for the problems represented with the contours having deep curved valleys.

In application of the axes rotation method to the optimization problems subject to nonlinear inequality constraints, an improved version of R.R. Allran and S.F.J. Johnsen's method is used, which deals with a new objective function composed of the original objective and a penalty term to consider the inequality constraints.

The program is incorporated in optimization code system SCOOP.

Keywords: ROTAX Code, Nonlinear Optimization, Axes Rotation Method,
Nonlinear Inequality Constraints, Orthogonal Coordinate,
Penalty Term, Optimization Code System SCOOP.

目 次

1. はじめに	1
2. 座標回転法のアルゴリズム	2
3. 非線形不等号制約条件式の扱い	5
4. テスト関数と計算結果	9
4.1 計算の条件	10
4.2 計算結果と収束効率の比較検討	11
5. 結論	25
謝辞	25
参考文献	25
付録1. ROTAXのFortranプログラムリスト	27
付録2. MODELプログラムのWildeの問題に対する例	30

1. はじめに

非線形最適化手法はそのアルゴリズムの型により傾斜法と直接探索法に分類することができる。それらの相異点は前者が各反復の過程で過去の情報を利用し、目的関数の値とその1階及び高階の導関数を計算して降下方向が反復毎に計算されるのに対し、後者は各反復の過程では過去の情報が用いられないことが多く、目的関数の値だけが計算され、探索方向は前もって決められている点にある。そして上述のような用いられる情報量の差から、概して傾斜法の方が収束効率の点で優れており精度の良い局所的な最小点（以後最適化問題を最小化問題と考えることにする）を求めるのに適した手法である。然しながら傾斜法は第4節の共役傾斜法、可変計量法、最急降下法、Newton法の例でみるように、その収束点が出発点の位置に依存し易いという欠点がある。これに対してパターン探索法、座標回転法、シンプレックス法や筆者が開発した球面探索法⁽¹²⁾などの直接探索法は収束効率の点では劣るが、与えられた点における目的関数の値だけが評価できれば良く任意の連続関数に対して用いることができる。また傾斜法に比べて出発点の位置にあまり左右されずに大域的な最小点に収束するという長所がある。このことは我々に傾斜法と直接探索法の使い方を示唆している。

上述の手法は制約条件を持たない最適化問題に対するものであるが、制約条件を持つ問題に対しても種々の手法が提案されている。例えばBox⁽³⁾により紹介されたコンプレックス法は、一つの頂点が制約条件の作る実行可能領域に含まれているようなコンプレックス（複体）を考え、これを求める最適点へと移動させて行く方法である。またCheneyとGoldstein⁽⁴⁾は出発点での近傍で制約条件を線形化し、線形計画法の問題として解く切除平面法を提唱している。線形計画法を用いるという点では他にGriffithとStewart⁽⁵⁾により提唱された近似計画法がある。これは目的関数と制約条件式を各反復の過程で得られる探索点の回りでTaylor展開し、その一次の項をとって線形化する方法である。AllranとJohnsen⁽⁶⁾は制約条件式を指数関数の中に組みこんだペナルティ項を考え、これをもとの目的関数に加えた新しい目的関数を作ってこの新しく作られた目的関数を通常の制約条件無しの最適化手法で解く方法を提唱した。この方法の利点は前述の制約条件無しの最適化手法をそのまま用いることができる点にある。

我々がここで紹介する最適化プログラムROTAXは、Rosenbrock⁽¹⁾により提案された制約条件無しの最適化問題に対する直接探索法の一つである座標回転法に、上記のAllranとJohnsen法を適用して制約条件式を持つ問題を解くために開発されたものである。座標回転法自体の手法の安定性と収束効率に対する検討が、Rosenbrockの関数、Bealeの関数、Boxの関数、Powellの関数、Woodの関数とEnzymeの関数をテスト関数としてなされた。またROTAXの制約条件を持つ問題に対する最適化プログラムとしての検討がRosen-Suznkiの問題、Bealeの問題とWildeの問題をテスト問題としてなされ、手法の安定性、収束効率、精度の点において優れた結果が得られた。

この研究は原子炉工学部内の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ある。

2. 座標回転法のアルゴリズム

Rosenbrock⁽¹⁾により提唱されたこの方法はHookeとJeeves⁽²⁾のパターン探索法をさらに発展させたものとみなすことができ、パターン探索法よりも効果的であるといわれている。サブルーチンROTAXはRosenbrockの方法に多少の修正を加えてコード化したもので、以下にアルゴリズムの概要を述べる。なお制約条件が有る場合についての解法の説明は次節にゆずり、ここでは制約条件無しの問題を考えるものとする。説明の前に次の量が与えられているものと仮定する。

- (i) $F(x)$: 変数 $x \in E^n$ の関数 (目的関数)
- (ii) x_0 : 変数 x の初期値
- (iii) $S = (s_1, s_2, \dots, s_n)$: 1組の互いに直交する n 個の方向。最初の S としては座標軸方向を用いる。
- (iv) α, β : 刻み幅を調節する係数で $\alpha > 1, 0 < \beta < 1$ とする。
- (v) $\lambda_0 = (\lambda_1^0, \lambda_2^0, \dots, \lambda_n^0)$: $s_j (j = 1, \dots, n)$ 方向に沿っての1組の刻み幅

アルゴリズム

- (i) $A_j = 0, \lambda_j = \lambda_j^0, j = 1, \dots, n$ とする。
- (ii) 出発点 x_0 から s_1 方向へ幅 λ_1 のステップをとる。
 即ち $x = x_0 + \lambda_1 s_1$ とし

$$F(x) \leq F(x_0)$$
 なら $A_1 = A_1 + \lambda_1, \lambda_1 = \alpha \lambda_1, x_1 = x$ とする。

$$F(x) > F(x_0)$$
 なら $\lambda_1 = -\beta \lambda_1, x_1 = x_0$ とする。

探索(ii)を全ての方向 $s_j, j = 1, \dots, n$ に対して行なう。(i), (ii)の探索を全ての方向のうち1つの方向でも目的関数が小さくなる方向がある限り続ける。全ての方向に対して失敗したら、即ち $A_j = 0 (j = 1, \dots, n)$ なら次のステップへ行く。

- (iii) 次の方向ベクトルの組を定義する。

$$Q = [q^1, q^2, \dots, q^n] = S \begin{pmatrix} A_1 & 0 & 0 & \dots & 0 \\ A_2 & A_2 & 0 & \dots & 0 \\ A_3 & A_3 & A_3 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ A_n & A_n & \dots & \dots & A_n \end{pmatrix} \quad (1)$$

(1)式で定義された方向の中で q^1 は出発点 x_0 と(i), (ii)の探索の結果到達した点 x_p を結ぶベクトルであり、 q^2 は s_1 方向以外の方向への成功したステップ幅の合計で、以下 q^3, \dots, q^n についても同様である。Fig. 1には $n = 3$ の場合についての q^1, q^2, q^3 が示されている。

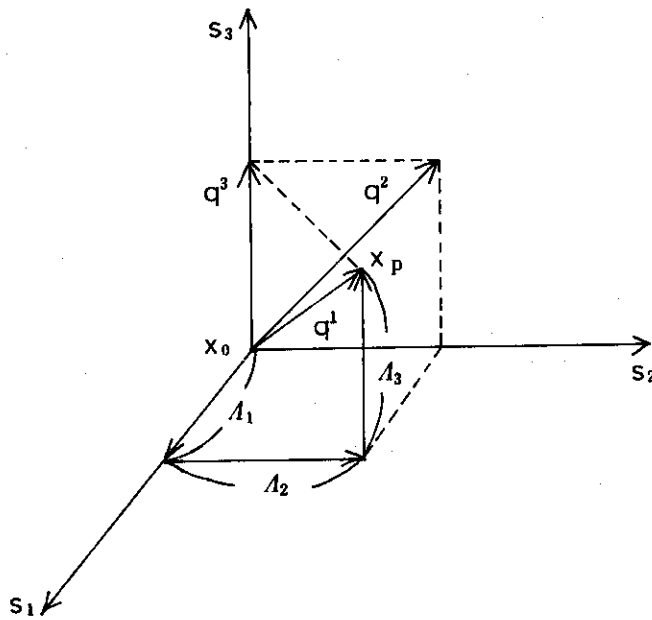


Fig. 1 Definition of the direction vector q for $n=3$

(V) (iii)で得られた方向ベクトルから Gram-Schmidt の直交化法を用いて 1 組の新しい直交ベクトルを次のように生成する。

$$\left. \begin{aligned} v^1 &= q^1 \\ v^i &= q^i - \sum_{j=1}^{i-1} [(q^i)^T s_j] s_j, \quad i = 2, \dots, n \end{aligned} \right\} \quad (2)$$

ここで

$$s_i = v^i / \|v^i\|, \quad i = 1, \dots, n$$

但し q^T はベクトル q の転置を意味し、 $\|\cdot\|$ はユークリッドのノルムを示す。これにより新しい方向 $\hat{S} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n)$ が作られる。

(i)から(V)までが 1 つの反復でくり返される過程である。また(2)式で計算される \hat{s}_1 方向はこの反復の段階での全てのステップ A_1, \dots, A_n により得られた方向である。Fig. 2には $n=2$ の場合について \hat{s}_1, \hat{s}_2 の生成の図が示されており、この \hat{s}_1 の方向が目的関数の最も減少する方向であると期待できる。

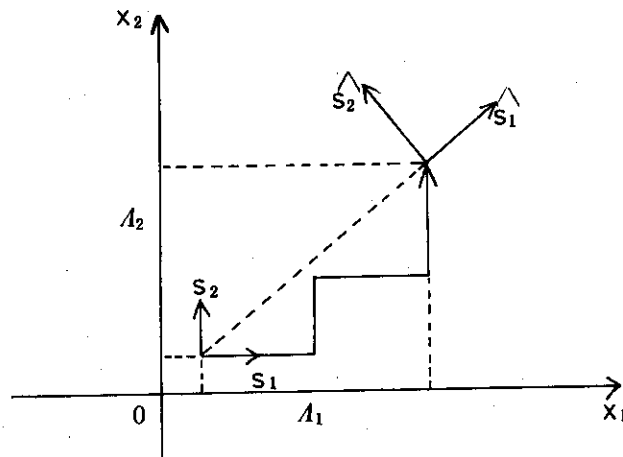


Fig. 2 Generation of new orthogonal coordinate system by axes rotation for $n=2$

- (V) 各方向に対する刻み幅 λ_j が許容誤差 ε_1 以下, 即ち $\max |\lambda_j| < \varepsilon_1$ になるか, ノルム $\|q^1\|$ があらかじめ定めた値以下, 即ち $\|q^1\| < \varepsilon_2$ になったら計算を終了する。そうでない場合は

$$\left. \begin{aligned} \lambda^0_j &:= \lambda^0_j \cdot e^{-ITR} \\ x_0 &:= x_p \\ S &:= \hat{S} = (\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n) \end{aligned} \right\} \quad (3)$$

として(i)へ戻る。ここで x_p は前の段階で到達した点であり, ITR は反復回数である。

(3)式の中で $\lambda^0_j = \lambda^0_j \cdot e^{-ITR}$ と縮少する理由は, 新しい直交座標系での目的関数の等高線は前の段階より狭くなっているだろうという仮定に基づくものである。

3. 非線形不等号制約条件式の扱い

ROTAXではユーザーの指示により制約条件を持つ問題を扱うことができるが、制約条件を持つ問題のために我々が用いた解法はAllranとJohnsenにより紹介されたもので、以下にその手法の概要を説明する。

制約条件を持つ非線形最適化問題は次のように定義することができる。

$$\left. \begin{aligned} &\text{minimize } F(x); x \in E^n \\ &\text{subject to } g_i(x) \geq 0, i = 1, \dots, m \end{aligned} \right\} \quad (4)$$

等号制約条件式は二つの不等号制約条件式により近似できるので、(4)式は非線形制約条件式を持つ最適化問題化に対する一般的な形と考えることができる。

補助関数 $F_\ell(x)$ を次の式で定義する。

$$F_\ell(x) = F(x) + \sum_{i=1}^m \exp [T_{i\ell} \cdot g_i(x)] \quad (5)$$

ここで列 $\{T_{i\ell}\}$ は各 $i = 1, \dots, m$ に対して

$$\left. \begin{aligned} &T_{i, \ell+1} < T_{i\ell} < 0, \\ &\lim_{\ell \rightarrow \infty} T_{i\ell} = -\infty \end{aligned} \right\} \quad (6)$$

なる性質を持つ数列である。

もし問題(4)が解を持ち、 F と g_i が $C^{(1)}$ 級で g_i の作る領域(実行可能領域)が空集合でなければ、各 $F_\ell(x)$ は各 ℓ に対して点 x^*_{ℓ} において唯一の最小値

$$\min_x F_\ell(x) = F_\ell(x^*_{\ell})$$

を持ち、十分大きな ℓ に対して $g_i(x^*_{\ell}) \geq 0, i = 1 \dots m$ である。即ち ℓ を十分大きくすると x^*_{ℓ} は実行可能領域に含まれる。更に実行可能領域内の x に対しては

$$\lim_{\ell \rightarrow \infty} [\min_x F_\ell(x)] = \inf_x F(x)$$

が成り立つ。即ち最小値 $\{F_\ell(x^*_{\ell})\}$ の列は問題(4)の解に収束する。

上述のアルゴリズムを直観的に見るために次の例を考えてみる (Fig. 3参照)

$$\left. \begin{aligned} &\text{minimize } F(x) = x \\ &\text{subject to } g(x) = x - 1 \geq 0 \end{aligned} \right\} \quad (7)$$

$T_\ell = -\ell$ をとると

$$F_\ell(x) = x + e^{-\ell(x-1)} \quad (8)$$

となり、各 ℓ に対して F_ℓ を最小にしながら $\ell \rightarrow \infty$ にすると(7)の解に収束するというのが上述のアルゴリズムの示すところである。実際(8)の最小値は

$$x^*_{\ell} = 1 + \frac{\log \ell}{\ell} \quad (9)$$

で与えられることが容易にわかり、(9)式で $\ell \rightarrow \infty$ とすると $\log \ell / \ell = 0$ となり $\lim_{\ell \rightarrow \infty} x^*_{\ell} = 1$ は

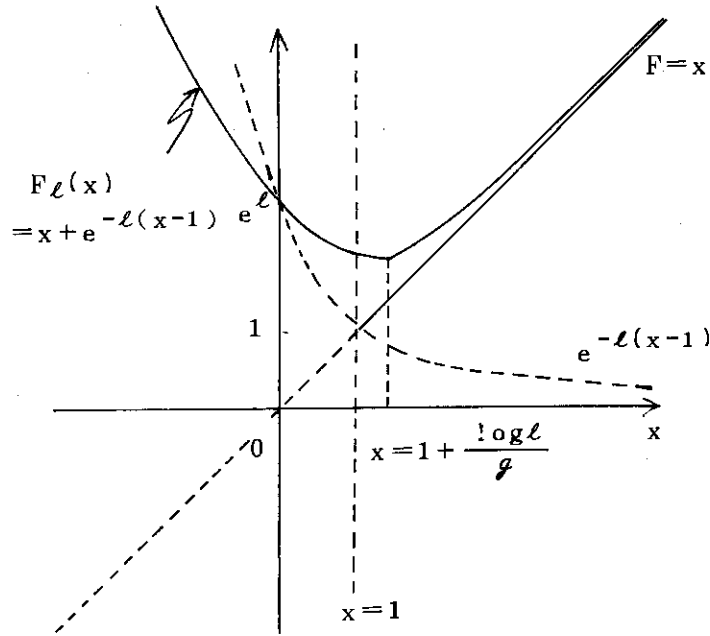


Fig. 3 Example of Allran and Johnsen's technique for the problem to find "min $F(x)=x$, subject to $g(x)=x-1 \geq 0$ "

(7)の解を与えている。 $T_{i\ell}$ の具体的な選び方は種々の形が考えられるが要は条件(6)が満たされるように選ぶことである。

AllranとJohnsenにより紹介された制約条件を持つ問題に対するアルゴリズムの概要は以上のとおりであるが、このアルゴリズムには若干の問題点がある。それは新しい目的関数が(5)式により作られ、各 ℓ 毎に $T_{i\ell}$ を定めて $F_\ell(x)$ を制約条件無しで最適化手法で最小化するわけであるが、このとき $T_{i\ell}$ を定める初期の段階で $F(x)$ の減少の仕方の方がペナルティ項の増え方をはるかに上回る場合である。例えば $F(x) = -e^{x^2}$ で $-1 \leq x \leq 1$ の範囲で最小値を求める問題を考えてみよう。(5)式に対応する新しい目的関数は

$$F_\ell(x) = -e^{x^2} + e^{T_{1\ell}(x+1)} + e^{T_{2\ell}(1-x)} \quad (10)$$

となり、 $T_{i\ell}$ として

$$T_{i\ell} = -\ell \quad (\ell = 1, 2, 3, \dots), \text{ for } i = 1, 2.$$

を選んだ場合を考えてみる。この選び方は条件(6)を満たしている。このとき $T_{i\ell}$ の初期の段階即ち $\ell = 1$ のとき(10)式は

$$F_1(x) = -e^{x^2} + e^{-(x+1)} + e^{(x-1)}$$

となり、 $g(x) = e^{-(x+1)} + e^{(x-1)}$ とするとFig. 4にみるように実行可能領域 (Fig. 4で斜線を施した部分)の外側での $F(x)$ の減少の仕方の方がペナルティ項 $g(x)$ の増加率を上回っている。この結果(10)式の最小化問題の解 x は実行可能領域をどんどんはみだして $\pm\infty$ に行ってしまう。このようなことが生じる原因は、ペナルティ項を(5)式で定義する限り、実行可能領

域の境界面上でのペナルティ項の値は1にしかならず，出発点が実行可能領域内にあってもその点から境界面への接近が容易なことで，境界の外においての $F(x)$ の減少率がペナルティ項の増加率を上回るときがあることにある。従ってこれを避けるために我々が用いたのは(5)式のペナルティ項に重みをつける方法である。即ち，

$$F_{\ell}(x) = F(x) + \sum_{i=1}^m W_i \exp [T_{i\ell} \cdot g_i(x)]$$

とし $W_i > 0$ を十分大きくとってやれば境界面への接近に対しては大きなペナルティが加えられることになる。

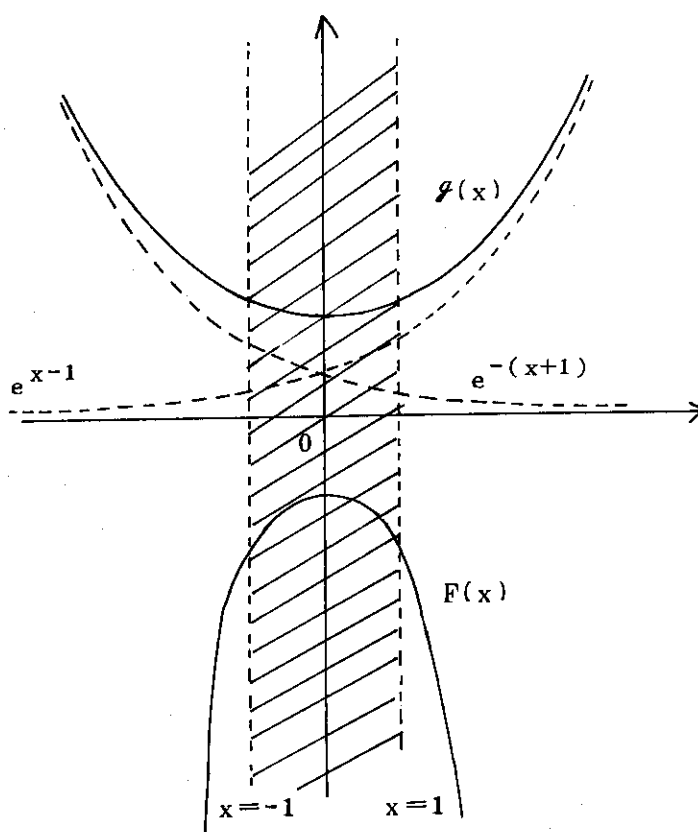


Fig. 4 One example of questions for Allran and Johnsen's technique by the problem to find "min $F(x) = -e^{x^2}$, subject to $-1 \leq x \leq 1$ "

Table 1.1, 2は上述の問題点を実際に ROTAXによる計算により示したもので， $T_{1\ell} = T_{2\ell} = -\ell^3$ ， $\ell = 1, 2, \dots, 10$ とし，出発点を $x = 0$ として計算した。Table 1aは重みを付けない場合の $\ell = 1$ に対する計算結果である。

表でみるように重みを付けない場合は $\ell = 1$ の段階で x は $F = -e^{x^2}$ に沿って実行可能領域を越えて $+\infty$ に行き計算がオーバーフローで打ち切られた。これに対し Table 1bは重み $W_1 = W_2 = 50.0$ を付けて計算した結果である。

表のように重みを付けた場合探索が常に実行可能領域内で行なわれ， $\ell = 10$ の段階でこの

問題の最適解 $x = 1$, $F = -2.7183$ の良い近似を与えている。

Table 1.1 Computational result of non-weighted Allran and Johnsen's technique by the problem to find " $\min F(x) = -e^{x^2}$; subject to $-1 \leq x \leq 1$ "

ℓ	ITR	x	$F_{\ell}(x)$
1	1	0.0	-0.2642
	2	0.01	-0.2643
	3	0.04	-0.2652
	4	0.13	-0.2751
	5	0.40	-0.3781
	6	1.21	-2.9803
	7	3.64	-0.5678+6
	8	10.93	-0.7637+52
	9	32.80	-0.2894+77
	10	98.41	"
	11	"	"
	12	"	"

Table 1.2 Computational result of weighted Allran and Johnsen's technique by the problem to find " $\min F(x) = -e^{x^2}$, subject to $-1 \leq x \leq 1$ "

ℓ	ITR	x	$F_{\ell}(x)$
1	4	0.0	35.7879
2	2	0.0	-0.9664
3	42	0.7711	-1.7088
4	18	0.8955	-2.1675
5	32	0.9422	-2.3932
6	11	0.9643	-2.5118
7	14	0.9762	-2.5791
8	12	0.9833	-2.6201
9	4	0.9833	-2.6297
10	4	0.9933	-2.6755

4. テスト関数と計算結果

サブルーチンROTAXの収束効率，安定性を調べるために用いたテスト関数は以下のとおりである。

〔I〕 制約条件無しの問題

(i) Rosenbrockの関数⁽¹⁾

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

この関数は $x_2 = x_1^2$ に沿って険しい谷があり， F は $x = (1, 1)$ で最小値 0 を持つ。

(ii) Bealeの関数⁽⁷⁾

$$F(x) = \sum_{i=1}^3 (C_i - x_1(1 - x_2^i))^2$$

ここに $C_1 = 1.5$, $C_2 = 2.25$, $C_3 = 2.625$ である。この関数は $x_2 = 1$ の近くに狭い曲がった谷を持ち， $x = (3, 0.5)$ で最小値 0 を持つ。

(iii) Boxの関数⁽⁸⁾

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

ここに $y_i = 0.1 \cdot i$, $i = 1, \dots, 10$ である。 F は $x = (1, 10, 1)$, $(10, 1, -1)$, $(x_1 \rightarrow +\infty, x_2 \rightarrow +\infty, x_3 = 0)$, $(x_1 = x_2, x_3 = 0)$ で最小値 0 を持つ。

(iv) Powellの関数⁽⁹⁾

$$F(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

F は点 $x = (0, 0, 0, 0)$ で最小値 0 を持つ。

(v) Woodの関数⁽⁹⁾

$$F(x) = 100(x_2 - x_1)^2 + (1 - x_1^2)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$$

F は点 $x = (1, 1, 1, 1)$ で最小値 0 を持つ。

(vi) Enzymeの関数⁽¹⁰⁾

$$F(x) = \sum_{i=1}^n (v_i - x_1(y_i^2 + x_2 y_i)) / (y_i^2 + x_3 y_i + x_4)^2$$

ここに 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 3.075×10^{-4} を持つ。

[II] 制約条件を持つ問題

(i) Rosen-Suzuki の問題⁽¹⁰⁾

$$\min F(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to

$$f_1 = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0$$

$$f_2 = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0$$

$$f_3 = -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0$$

Fは $x = (0, 1, 2, -1)$ で最小値 -44 をとる。

(ii) Beale の問題⁽¹⁰⁾

$$\min F(x) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to

$$f_1(x) = x_1 \geq 0, \quad f_2(x) = x_2 \geq 0, \quad f_3(x) = x_3 \geq 0,$$

$$f_4(x) = 3 - x_1 - x_2 - 2x_3 \geq 0$$

Fは $x = (4/3, 7/9, 4/9)$ で最小値 $1/9$ をとる。

(iii) Wilde の問題⁽¹¹⁾

$$\min F(x) = -e^{-(x_1 - 1)^2 + (x_2 - 1)^2}$$

subject to

$$f_1(x) = x_1 \geq 0, \quad f_2(x) = x_2 \geq 0$$

$$f_3(x) = x_1 - x_2^2 \geq 0$$

$$f_4(x) = -e^{-x_1} + x_2 \geq 0$$

$$f_5(x) = -2(x_1 - 1)^2 + x_2 \geq 0$$

Fは $x = (1.3586, 0.2571)$ で最小値 -2.38 をとる。

4.1 計算の条件

計算に際し与えた出発点は、[I]に対しては以下の範囲で10点づつランダムに発生させた。

問題(i), (ii)に対しては $x_1, x_2 \in [-100, 100]$

問題(iii)に対しては $x_1 \in [-10, 10], x_2 \in [0, 20], x_3 \in [-100, 100]$

問題(IV), (V)に対しては $x_1, x_2, x_3, x_4, x_5 \in [-100, 100]$

問題(VI)に対しては $x_1, x_2, x_3, x_4 \in [-1, 1]$

また [II]に対しては問題(i), (ii), (iii)に対しそれぞれ $x_0 = (0, 0, 0, 0), (0, 0, 0), (1, 1)$ を出発点とした。また各問題に対して定めた刻み幅 λ_0 , 刻み幅調節係数 α, β 及び収束判定量 ϵ_1, ϵ_2 は Table 2 に示されている。

4.2 計算結果と収束効率の比較検討

制約条件が無い場合の最適化プログラムとしてのROTAXの計算効率及び手法の安定性を他のプログラムと比較するため用いたのは、最適化コードシステムSCOOPの中の共役傾斜法によるプログラムVA08AとCGD, 可変計量法によるプログラムFPD, 最急降下法とNewton法によるプログラムMINIMの4個のサブルーチンである。それぞれのサブルーチンで(i)から(VI)までの問題を解き、各問題に対してそれら4個のサブルーチンの中で収束効率, 安定性の面で最も良かったサブルーチンの結果と比較した。

なお収束効率C.E.は次の式で定義した。

$$C.E. = \ell n \frac{|F_0 - F_m|}{|FF - F_m|} / T \quad (1)$$

- ここに 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 (msec)

制約条件を持つ問題 [II]に対してはその計算結果をのせるにとどめる。

(i) Rosenbrockの問題

問題に対して作られた10組の出発点は次のとおりであり、各出発点に対するROTAXの計算結果と他のサブルーチンの中では最も良かったFPDの結果を Table 3 に示す。

Run %	1	2	3	4	5	6	7	8	9	10
x_1	63.67	-86.03	135.3	27.85	48.64	-4.22	74.57	78.88	23.25	-8.31
x_2	33.37	20.63	-18.14	16.69	-21.7	79.53	64.41	-82.36	-39.07	0.49

Table 3 の中でRun%は10個の出発点の組に対応し、ITNは収束に要した反復回数, IFNは関数値計算プログラムの呼び出し回数, F-finalは収束点における目的関数の値, X-finalは収束した点, Tは要した計算時間である。なおROTAXの計算結果でIFNの欄が空白になっているがこれはIFNがITNから計算できるからである。即ち変数の数をnとすると $IFN = ITN \times (n+1)$ である。

表のように全ての出発点に対しROTAX, FPDとも最適点に収束している。

Table 4 はRosenbrockの問題に対するROTAX, FPDの収束効率C.E.を(1)式から求めたものである。

Table 2.1 Fixed parameters α , β , λ_0 , ε_1 and ε_2 for
Rosenbrock's and Beale's problems

Rosenbrock								
Run No	α	β	λ_0				ε_1	ε_2
1	3.0	0.5	0.01	0.01			10^{-7}	10^{-7}
2	2.0	0.5	0.005	0.005			"	"
3	2.0	0.5	0.005	0.005			"	"
4	3.0	0.5	0.008	0.008			"	"
5	1.5	0.5	0.005	0.005			"	"
6	2.0	0.5	0.005	0.005			"	"
7	"	"	"	"			"	"
8	"	"	"	"			"	"
9	1.5	0.5	"	"			"	"
10	2.0	0.5	"	"			"	"
Beale								
1	2.0	0.5	0.1	0.1			10^{-7}	10^{-7}
2	1.5	0.2	0.01	0.01			"	"
3	2.0	0.5	0.1	0.1			"	"
4	"	"	"	"			"	"
5	"	"	0.01	0.01			"	"
6	"	"	"	"			"	"
7	"	"	0.1	0.1			"	"
8	1.5	0.2	0.01	0.01			"	"
9	2.0	0.5	"	"			"	"
10	3.0	0.5	0.1	0.1			"	"

Table 2.2 Fixed parameters α , β , λ_0 , ϵ_1 and ϵ_2 for
Box's and Powell's problems

Box								
Run No.	α	β	λ_0				ϵ_1	ϵ_2
1	2.0	0.5	0.05	0.01	0.05		10^{-7}	10^{-7}
2	"	"	0.05	0.05	0.05		"	"
3	"	"	"	"	"		"	"
4	"	"	"	"	"		"	"
5	"	"	0.05	0.001	0.05		"	"
6	"	"	0.05	0.05	0.05		"	"
7	"	"	"	"	"		"	"
8	"	"	0.05	0.001	0.05		"	"
9	"	"	0.05	0.005	0.05		"	"
10	"	"	0.01	0.01	0.01		"	"
Powell								
1	3.0	0.5	0.1	0.1	0.1	0.1	10^{-7}	10^{-7}
2	"	"	"	"	"	"	"	"
3	"	"	"	"	"	"	"	"
4	"	"	"	"	"	"	"	"
5	"	"	"	"	"	"	"	"
6	"	"	"	"	"	"	"	"
7	"	"	"	"	"	"	"	"
8	"	"	"	"	"	"	"	"
9	"	"	"	"	"	"	"	"
10	"	"	"	"	"	"	"	"

Table 23 Fixed parameters α , β , λ_0 , ϵ_1 and ϵ_2 for
Wood's and Enzyme's problems

Wood								
Run No.	α	β	λ_0				ϵ_1	ϵ_2
1	3.0	0.5	0.05	0.05	0.05	0.05	10^{-7}	10^{-7}
2	"	"	"	"	"	"	"	"
3	"	"	0.7	0.8	0.5	0.5	"	"
4	"	"	0.05	0.05	0.05	0.05	"	"
5	"	"	"	"	"	"	"	"
6	"	"	"	"	"	"	"	"
7	"	"	0.6	0.5	0.6	0.7	"	"
8	"	"	0.05	0.05	0.05	0.05	"	"
9	"	"	"	"	"	"	"	"
10	"	"	"	"	"	"	"	"
Enzyme								
1	2.0	0.5	0.05	0.05	0.05	0.05	10^{-7}	10^{-7}
2	"	"	0.05	0.05	0.05	0.2	"	"
3	"	"	"	"	"	1.0	"	"
4	"	"	"	"	"	"	"	"
5	"	"	"	"	"	0.2	"	"
6	"	"	"	"	"	"	"	"
7	"	"	"	"	"	0.5	"	"
8	"	"	"	"	"	1.0	"	"
9	"	"	"	"	"	0.5	"	"
10	"	"	"	"	"	0.5	"	"

Table 3 Computational result for Rosenbrock's function
by ROTAX and FPD

Program	Run №	ITN	IFN	F-final	× - final	T(sec)
ROTAX	1	2994		0.2165-6	0.9996, 0.9992	0.240
	2	432		0.1580-6	1.0, 1.0	0.035
	3	2210		0.0	"	0.170
	4	1407		0.3976-9	"	0.115
	5	3155		0.8082-10	"	0.246
	6	3146		0.2132-11	"	0.248
	7	496		0.1110-12	"	0.045
	8	259		0.7208-7	0.9998, 0.9995	0.021
	9	258		0.2492-11	1.0, 1.0	0.021
	10	10128		0.4798-11	"	0.753
FPD	1	38	184	0.5121-29	1.0, 1.0	0.284
	2	41	208	0.0	"	0.294
	3	20	69	0.4933-29	"	0.140
	4	40	196	0.0	"	0.285
	5	22	109	0.2645-29	"	0.172
	6	55	313	0.0	"	0.404
	7	49	254	0.0	"	0.335
	8	18	81	0.1536-26	"	0.131
	9	14	125	0.2091-30	"	0.113
	10	16	107	0.2111-30	"	0.125

Table 4 Convergence efficiency of ROTAX and FPD for
Rosenbrock's problem

Run № \ Program	1	2	3	4	5	6	7	8	9	10
ROTAX	1.523	13.511	7.532	3.436	1.764	1.602	11.479	18.356	20.974	0.526
FPD	3.121	6.480	5.907	6.525	5.132	4.479	5.669	6.401	7.785	6.698

これからわかることはFPDの収束効率は全ての出発点に対して分散が少ないのに対しROTAXでは非常に変動が大きく、とくにRun No 2, 7, 8, 9に対しては良い効率を示した反面Run No 10に対しては悪かった。

(ii) Bealeの問題

10組の出発点はRosenbrokの問題と同じである。Table 5はこの問題に対するROTAXの結果と他のサブルーチンの中では最も良かったMINIMの計算結果である。

ROTAXでは全ての出発点に対して最適点に収束しているが、MINIMではRun No 2, 5, 6, 9が $x_2=1$ に沿った谷の中に落ちこんでしまっている。なおRun Noの左側に印した×印はそのRun Noの計算は最適点へ収束しなかったことを意味する。

Table 6はBealeの問題に対するROTAX, MINIMの収束効率である。

表のように全ての出発点に対してROTAXの方が良い収束効率を示しており、Bealeの問題に対しては安定性の面と収束効率の面でROTAXが優れていることがわかる。

(iii) Boxの問題

出発点は次のとおりである。

No	1	2	3	4	5	6	7	8	9	10
x_1	6.37	2.06	-3.06	-5.36	3.78	-8.34	9.31	-5.89	3.38	-4.99
x_2	13.34	19.31	7.97	13.36	3.71	5.58	4.39	1.64	3.61	5.25
x_3	27.17	2.77	-46.24	84.00	82.50	85.57	25.57	28.52	-87.45	-35.13

この問題に対するROTAXの計算結果と他の4個のサブルーチンの中では最も良かったCGDの結果がTable 7に示されている。

表のように全ての出発点に対してROTAX, CGDとも最適点に収束している。但しROTAXのRun No 3, 6, 9, 10とCGDのRun No 1, 5, 7, 8, 9が $x_1=x_2, x_3 \rightarrow 0$ 即ち $x_1 \cdot x_2$ 平面で直線 $x_1=x_2$ 上に、またROTAXのRun No 4, 7とCGDのRun No 3, 10が $x_1 \cdot x_2 \rightarrow +\infty, x_3 \rightarrow 0$ なる点に収束している。

Table 8はBoxの問題に対するROTAXとCGDの収束効率である。

収束効率の点ではROTAX, CGDともほぼ同程度である。

(iv) Powellの問題

10組の出発点は以下のとおりである。

No	1	2	3	4	5	6	7	8	9	10
x_1	63.67	98.07	-76.24	1.07	22.34	-28.73	32.36	-19.78	38.98	41.66
x_2	33.37	2.77	-95.18	50.46	-13.29	-25.28	-32.05	90.23	13.97	38.22
x_3	27.17	-65.99	45.02	93.08	-80.78	-93.11	-83.56	-80.57	-66.95	54.43
x_4	62.67	-20.90	-73.23	-95.63	34.91	34.91	-12.88	-95.49	72.49	-17.46

Table 9はこの問題に対するROTAXの計算結果と他のサブルーチンの中で最も良か

ったMINIMの結果である。

ROTAX, MINIMともに全ての出発点に対して最適点へ収束しており、次のTable 10に見るとおり収束効率も良く、この関数の性質はいわゆるおとなしい関数であることを示している。

(V) Wood の問題

出発点はPowellの問題と同じである。Table 11はこの問題に対するROTAXの計算結果と他のサブルーチンの中で最も良かったMINIMの結果である。

ROTAX, MINIAとも全て最適点に収束している。Table 12はこの問題に対する収束効率である。収束効率の点ではROTAXの方が優れているのがわかる。

(VI) Enzyme の問題

10組の出発点はPowellの問題, Woodの問題の出発点の各座標を1/100にした点である。ROTAXによる計算結果と他のルーチンの中で良かったVA08Aの結果がTable 13に示されている。

ROTAXでは全ての出発点に対してほぼ1%程度のずれはあるが最適点に収束しているのがわかる。これに対しVA08AではRun No. 3, 4, 8, 10に対しては最適点に収束しなかった。なお他のサブルーチンの場合, MINIMではRun No. 2, 3, 5, 7, 8, 9, 10の6点に対し, CGDでは全ての出発点に対し, FPDではRun No. 2, 3, 4, 6, 7, 8, 10の7点に対し最適点に収束しておらず, これらのサブルーチンの比較の上ではROTAXが非常に安定したサブルーチンであることを示している。Table 14にはこの問題に対するROTAXとVA08Aの収束効率が示されている。

表に見るように収束効率の点でもROTAXは優れていることがわかる。

制約条件を持つ問題の計算結果

(i) Rosen-Suzuki の問題

重み W_i 及び $T_{i\ell}$ を次のように定めた。

$$\left. \begin{array}{l} W_i = 1 \\ T_{i\ell} = -\ell^3, (\ell = 1, \dots, 10) \end{array} \right\} \text{ for } i = 1, 2, 3$$

Table 15はこの問題に対する計算結果を各 ℓ 毎に示したもので、ITR, F, \times はそれぞれ各段階でのくり返し回数, その段階での最終的な目的関数の値と変数の値である。またTは累積計算時間である。

(ii) Bealeの問題

重み W_i と $T_{i\ell}$ は Rosen-Suzuki の問題と同様に定め, $\ell = 1, \dots, 10$ に対しての計算結果をTable 16に示す。

(iii) Wildeの問題

重み W_i 及び $T_{i\ell}$ を次のように定める。

$$\left. \begin{array}{l} W_i = 5.0 \\ T_{i\ell} = -\ell^3, (\ell = 1, 2, \dots, 10) \end{array} \right\} \text{ for } i = 1, 2, 3, 4, 5$$

Table 17はこの問題に対する計算結果である。

Fig. 5は上の計算結果から探索の過程を図示したもので図の斜線部分が制約条件式の作る実行可能領域である。

Table 5 Computational result for Beale's function
by ROTAX and MINIM

Program	Run No	ITN	IFN	F-final	⊗ - final	T(sec)
ROTAX	1	264		0.5121-14	3.0, 0.5	0.027
	2	2916		0.7230-14	"	0.263
	3	104		0.3836-11	"	0.011
	4	116		0.5690-15	"	0.013
	5	242		0.5577-11	"	0.024
	6	172		0.5046-13	"	0.017
	7	301		0.2563-11	"	0.030
	8	198		0.5121-10	"	0.020
	9	112		0.1350-14	"	0.012
	10	147		0.7906-12	"	0.016
MINIM	1	11	33	0.7352-24	3.0, 0.5	0.099
	2	596	845	0.452	-0.186+7, 1.0	4.08
	3	8	19	0.1817-20	3.0, 0.5	0.061
	4	24	49	0.5640-26	"	0.168
	5	474	667	0.452	-0.978+5, 1.0	3.143
	6	577	807	0.452	-0.120+6, 1.0	3.904
	7	12	36	0.5683-28	3.0, 0.5	0.089
	8	13	46	0.1491-17	"	0.098
	9	445	628	0.452	-0.748+5, 1.0	2.965
	10	10	19	0.4615-23	3.0, 0.5	0.074

Mark × means the computation has no convergence for the Run No

Table 6 Convergence efficiency of ROTAX and MINIM
for Beale's problem

Run No \ Program	1	2	3	4	5	6	7	8	9	10
ROTAX	23095	2270	44445	45177	21725	35.199	20.129	35.554	44.428	21.175
MINIM	8.578	—	11.534	4.999	—	—	11.085	7.781	—	8.074

Table 7 Computational result for Box's function by
ROTAX and CGD

Program	Run №	ITN	IFN	F-final	× - final	T(sec)
ROTAX	1	352		0.1143-14	1.0, 10.0, 1.0	0.878
	2	204		0.1458-11	"	0.511
	3	165		0.4385-9	53.19, 53.24, 0.37-5	0.414
	4	171		0.4606-52	0.56+42, 0.15+51, -0.39-26	0.365
	5	406		0.2175-13	1.0, 10.0, 1.0	1.014
	6	179		0.1072-10	36.68, 36.68, -0.68-6	0.449
	7	42		0.1672-6	77.15, 109.0, 0.52-4	0.106
	8	243		0.8046-12	1.0, 10.0, 1.0	0.608
	9	251		0.5069-14	8051, 8051, -0.55-8	0.629
	10	156		0.1604-14	11.16, 11.16, -0.13-7	0.391
CGD	1	11	38	0.9648-32	(10.67, 10.67, -0.38-16)	0.334
	2	44	101	0.5826-22	(1, 10, 1)	0.947
	3	37	123	0.2671-11	(535, 133, -0.3-6)	0.865
	4	44	106	0.1064-21	(1, 10, 1)	0.868
	5	15	59	0.3416-32	(9.24, 9.24, -0.29-16)	0.429
	6	46	112	0.6391-22	(1, 10, 1)	0.883
	7	7	22	0.5891-19	(787, 787, 0.16-9)	0.164
	8	15	47	0.8341-33	(5.66, 5.66, 0.114-96)	0.351
	9	15	59	0.4824-32	(11.28, 11.28, -0.84-18)	0.445
	10	41	137	0.3332-11	(4815, 1316, -0.34-6)	0.892

Table 8 Convergence efficiency of ROTAX and CGD
for Box's problem

Run № \ Program	1	2	3	4	5	6	7	8	9	10
ROTAX	0.408	0.584	0.745	3.600	0.408	0.938	2.191	0.656	0.683	1.158
CGD	2.441	0.573	0.417	0.710	1.972	0.774	3.161	2.508	1.920	0.425

Table 9 Computational result for Powell's function
by ROTAX and MINIM

Program	Run%	ITN	IFN	F-final	× - final	T(sec)
ROTAX	1	3400		0.8450-15	-0.14-3, -0.14-4, -0.66-4, -0.66-4	0.595
	2	999		0.7498-12	0.78-3, -0.78-4, 0.38-3, 0.38-3	0.201
	3	719		0.5653-17	-0.41-4, 0.41-5, -0.17-4, -0.17-4	0.145
	4	440		0.8359-10	0.26-2, -0.26-3, 0.11-2, 0.11-2	0.091
	5	229		0.6310-8	0.65-2, -0.65-3, 0.15-2, 0.15-2	0.050
	6	619		0.4892-14	0.1-3, -0.1-4, -0.42-4, -0.42-4	0.127
	7	753		0.3164-10	-0.49-3, 0.49-4, -0.11-2, -0.11-2	0.152
	8	1849		0.9168-12	0.39-3, -0.39-4, 0.4-3, 0.4-3	0.326
	9	1177		0.1936-10	0.17-2, -0.17-3, 0.66-3, 0.66-3	0.235
	10	599		0.1781-8	0.54-2, -0.54-3, 0.2-2, 0.2-2	0.125
MINIM	1	25	26	0.4751-12	0.43-3, -0.43-4, 0.04, 0.04	0.279
	2	31	32	0.3409-12	0.017, -0.0017, -0.024, -0.024	0.326
	3	30	31	0.8744-12	0.59-3, -0.59-4, 0.45-3, 0.45-3	0.309
	4	30	31	0.8978-12	0.081, -0.0081, 0.03-2, 0.03-2	0.313
	5	30	31	0.3578-12	-0.43-3, 0.43-4, -0.36-3, -0.36-3	0.307
	6	30	31	0.1166-11	-0.88-3, 0.88-4, -0.37-3, -0.37-3	0.305
	7	30	31	0.2771-12	-0.011, 0.11-2, -0.034, -0.034	0.308
	8	31	32	0.6313-12	-0.016, 0.16-2, -0.035, -0.035	0.309
	9	30	31	0.3631-12	-0.053, 0.53-2, -0.035, -0.035	0.307
	10	29	30	0.5505-12	0.07, -0.7-2, 0.024, 0.024	0.296

Table 10 Convergence efficiency of ROTAX and MINIM

Run % Program	1	2	3	4	5	6	7	8	9	10
ROTAX	0.798	2.462	4.180	4.848	7.775	4.262	2.889	1.531	1.901	3.116
MINIM	1.475	1.401	1.426	1.407	1.435	1.444	1.430	1.478	1.435	1.433

Table 11 Computational result for Wood's function
by ROTAX and MINIM

Program	Run No	ITN	IFN	F-final	× - final	T(sec)
ROTAX	1	659		0.1791-11	1.0, 1.0, 1.0, 1.0	0.150
	2	1170		0.3784-10	" , 0.99999	0.257
	3	1633		0.8113-13	1.0, 1.0, 1.0, 1.0	0.372
	4	646		0.2695-11	"	0.148
	5	1386		0.5595-13	"	0.312
	6	319		0.9240-7	0.9999, 0.9997, 1.0, 1.0	0.075
	7	1157		0.2585-10	1.0, 1.0, 1.0, 1.0	0.262
	8	1204		0.4699-12	"	0.275
	9	1424		0.2566-9	1.0, 1.0, 0.99999, 0.99998	0.322
	10	1071		0.1068-12	1.0, 1.0, 1.0, 1.0	0.243
MINIM	1	29	31	0.2906-18	1.0, 1.0, 1.0, 1.0	0.351
	2	29	32	0.2470-18	"	0.350
	3	35	40	0.1859-16	"	0.386
	4	28	52	0.1741-16	"	0.309
	5	40	47	0.7514-18	"	0.443
	6	40	46	0.1242-16	"	0.424
	7	40	46	0.2052-17	"	0.422
	8	41	49	0.2382-18	"	0.452
	9	39	45	0.3939-17	"	0.431
	10	26	27	0.2585-16	"	0.283

Table 12 Convergence efficiency of ROTAX and MINIM
for Wood's problem

Run No \ Program	1	2	3	4	5	6	7	8	9	10
ROTAX	3218	1833	13364	3331	1685	5,177	1,779	1835	1,350	2085
MINIM	1821	1885	1,532	1,982	1,440	1,452	1,437	1,437	1,426	2085

Table 13 Computational result for Enzyme's function
by ROTAX and VA08A

Program	Run №	ITN	IFN	F-final	× - final	T(sec)	
ROTAX	1	529		03106-3	0.1924, 0.2049, 0.1280, 0.1419	0.328	
	2	498		03112-3	0.1929, 0.2068, 0.1357, 0.1414	0.310	
	3	157		03104-3	0.1927, 0.1958, 0.1252, 0.1378	0.101	
	4	256		03107-3	0.1925, 0.1891, 0.1182, 0.1355	0.163	
	5	89		03104-3	0.1927, 0.1941, 0.1240, 0.1372	0.060	
	6	156		03104-3	0.1927, 0.1932, 0.1234, 0.1368	0.100	
	7	75		03104-3	0.1927, 0.1946, 0.1249, 0.1372	0.049	
	8	294		03111-3	0.1931, 0.2034, 0.1351, 0.1403	0.184	
	9	165		03104-3	0.1931, 0.1885, 0.1248, 0.1344	0.107	
	10	372		03104-3	0.1927, 0.1952, 0.1245, 0.1376	0.233	
VA08A	1	129	306	03075-3	0.1928, 0.1909, 0.1231, 0.1358	1.712	
	2	93	235	03075-3	"	1.180	
	×	3	80	229	04242-3	0.2253, -0.415, -0.024, -0.178	1.102
	×	4	145	443	04242-3	"	2.056
	5	63	156	03075-3	0.1928, 0.1909, 0.1231, 0.1358	0.803	
	6	78	203	03075-3	"	1.011	
	7	49	119	03075-3	"	0.645	
	×	8	154	372	01594-2	0.234, -1.292, -0.836, -0.55	1.915
	9	132	322	03075-3	0.1928, 0.1909, 0.1231, 0.1358	1.644	
	×	10	401	1446	09635-3	-0.057, -55.85, 9.88, 6.57	5.844

Table 14 Convergence efficiency of ROTAX and
VA08A for Enzyme's problem

Run № \ Program	1	2	3	4	5	6	7	8	9	10
ROTAX	0.036	0.054	0.125	0.066	0.167	0.126	0.333	0.070	0.105	0.064
VA08A	0.069	0.099			0.137	0.111	0.180		0.068	

Table 15 Computational result for Rosen-Suzuki's
Problem by ROTAX

ℓ	ITR	F	\times	T(sec)
1	64	-41.946	0.108, 0.954, 2.086, -0.851	0.031
2	59	-43.011	-0.008, 0.991, 1.971, -0.967	0.062
3	1287	-43.572	-0.007, 0.996, 1.983, -0.991	0.663
4	50	-43.778	-0.007, 0.996, 1.992, -0.994	0.691
5	50	-43.871	-0.0008, 0.997, 1.993, -1.0003	0.719
6	10	-43.917	-0.0008, 1.007, 1.993, -1.0014	0.725
7	590	-43.944	0.0008, 0.9966, 1.996, -1.0012	0.991
8	2	-43.952	0.0008, 0.9966, 1.996, -1.0012	0.994
9	42	-43.977	0.0005, 1.005, 1.997, -1.0017	1.017
10	2	-43.980	0.0005, 1.005, 1.997, -1.0017	1.020

Table 16 Computational result for Beale's problem
by ROTAX

ℓ	ITR	F	\times	T(sec)
1	593	2.4077	1.3625, 0.7539, 0.2841	0.227
2	43	0.1986	1.3983, 0.7184, 0.3202	0.246
3	1080	0.1306	1.3573, 0.7618, 0.4044	0.657
4	117	0.1182	1.3425, 0.7716, 0.4292	0.701
5	88	0.1143	1.3377, 0.7749, 0.4373	0.733
6	2	0.1139	1.3377, 0.7749, 0.4373	0.735
7	318	0.1120	1.3346, 0.7769, 0.4423	0.840
8	2	0.1119	1.3346, 0.7769, 0.4423	0.842
9	4	0.1119	1.3346, 0.7769, 0.4423	0.846
10	2	0.1119	1.3346, 0.7769, 0.4423	0.851

Table 17 Computational result for Wilde's problem
by ROTAX

ℓ	ITR	F	\times	T(sec)
1	57	109.498	1.4595, 0.9991	0.022
2	42	-3.355	1.3198, 0.7486	0.040
3	53	-12.797	1.3387, 0.3998	0.062
4	47	-17.652	1.3498, 0.3260	0.082
5	33	-20.061	1.3540, 0.2967	0.097
6	28	-21.354	1.3559, 0.2822	0.109
7	38	-22.102	1.3568, 0.2742	0.125
8	5	-22.561	1.3568, 0.2692	0.128
9	7	-23.048	1.3568, 0.2642	0.133
10	2	-23.107	1.3568, 0.2642	0.135

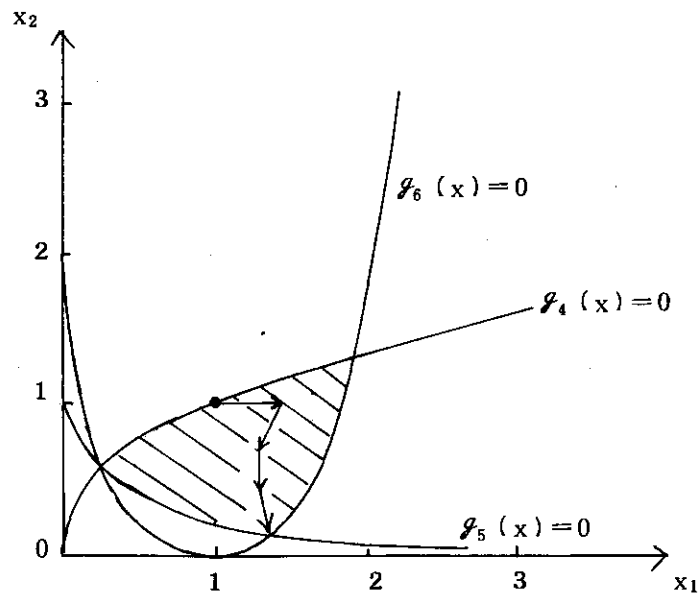


Fig. 5 Trajectory of searching point for weighted
Allran and Johnsen's technique for Wilde's
problem

5. 結 論

計算結果が示すように座標回転法は出発点に依存せずに大域的な最小点へ収束するという手法の安定性においてだけでなく、収束効率の点でも優れた結果を与えており、最適化コードシステムSCOOPの中でも中心的な手法になるものと期待される。しかしシンプレックス法やパターン探索法などの他の直接探索法がそうであるように、手法に固有の係数即ち刻み幅 λ_0 、刻み幅調節係数 α 、 β の決定がその効率に大きな影響を与える反面それらの定め方の基準が与えられていないという欠点がある。テスト関数として用いた Rosenbrock, Beale, Box, Powell, Wood と Enzyme の関数に対して我々が定めた値は Table 2.1 ~ 2.3 に示されているが、ある程度の試行錯誤的な決定に頼らざるを得ないという難しさが残されている。我々の使用経験から言えば、刻み幅の決定はそれに対応する変数の目的関数に対する感度 (sensitivity) によりなされるべきであると思われる。即ち感度の高い変数に対する刻み幅は小さく定め、逆に低い変数に対しては多少大きめに定めることが望ましい。また調節係数 α はほぼ 2.0 ~ 3.0 程度、 β は 0.5 が妥当と思われる。いずれにしてもこれらの量の決定法は今後の課題であろう。

制約条件を持つ問題に対する ROTAX の精度は 3 つのテスト問題でみる限り満足のいく解が得られており、Allran と Johnsen 法を他の最適化手法へ適用する 1 つの保証が得られた。重み W_i は境界面 $f_i(x) = 0$ 上での目的関数の値により決定されるべきもので、境界面上での関数値が小さい場合には大きく定める必要がある。

謝 辞

本プログラムの開発にあたり終始貴重な御意見と手法上の検討をいただいた篠原慶邦主任研究員をはじめとするシステム解析手法研究グループの皆さんに心から感謝の意を表します。

参 考 文 献

- (1) Rosenbrock H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3, 175 (1960)
- (2) Hooke R. & Jeeves T.A., "Direct Search Solution of Numerical & Statistical Problems", J. Assoc. Comput. Math., 8, 212 (1961)
- (3) Box M. J., "A New Method of Constrained Optimization and a Comparison with other Methods", Computer J., 8, 42 (1965)

5. 結 論

計算結果が示すように座標回転法は出発点に依存せずに大域的な最小点へ収束するという手法の安定性においてだけでなく、収束効率の点でも優れた結果を与えており、最適化コードシステムSCOOPの中でも中心的な手法になるものと期待される。しかしシンプレックス法やパターン探索法などの他の直接探索法がそうであるように、手法に固有の係数即ち刻み幅 λ_0 、刻み幅調節係数 α 、 β の決定がその効率に大きな影響を与える反面それらの定め方の基準が与えられていないという欠点がある。テスト関数として用いた Rosenbrock, Beale, Box, Powell, Wood と Enzyme の関数に対して我々が定めた値は Table 2.1 ~ 2.3 に示されているが、ある程度の試行錯誤的な決定に頼らざるを得ないという難しさが残されている。我々の使用経験から言えば、刻み幅の決定はそれに対応する変数の目的関数に対する感度 (sensitivity) によりなされるべきであると思われる。即ち感度の高い変数に対する刻み幅は小さく定め、逆に低い変数に対しては多少大きめに定めることが望ましい。また調節係数 α はほぼ 2.0 ~ 3.0 程度、 β は 0.5 が妥当と思われる。いずれにしてもこれらの量の決定法は今後の課題であろう。

制約条件を持つ問題に対する ROTAX の精度は 3 つのテスト問題でみる限り満足のいく解が得られており、Allran と Johnsen 法を他の最適化手法へ適用する 1 つの保証が得られた。重み W_i は境界面 $f_j(x) = 0$ 上での目的関数の値により決定されるべきもので、境界面上での関数値が小さい場合には大きく定める必要がある。

謝 辞

本プログラムの開発にあたり終始貴重な御意見と手法上の検討をいただいた篠原慶邦主任研究員をはじめとするシステム解析手法研究グループの皆さんに心から感謝の意を表します。

参 考 文 献

- (1) Rosenbrock H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3, 175 (1960)
- (2) Hooke R. & Jeeves T.A., "Direct Search Solution of Numerical & Statistical Problems", J. Assoc. Comput. Math., 8, 212 (1961)
- (3) Box M.J., "A New Method of Constrained Optimization and a Comparison with other Methods", Computer J., 8, 42 (1965)

5. 結 論

計算結果が示すように座標回転法は出発点に依存せずに大域的な最小点へ収束するという手法の安定性においてだけでなく、収束効率の点でも優れた結果を与えており、最適化コードシステムSCOOPの中でも中心的な手法になるものと期待される。しかしシンプレックス法やパターン探索法などの他の直接探索法がそうであるように、手法に固有の係数即ち刻み幅 λ_0 、刻み幅調節係数 α 、 β の決定がその効率に大きな影響を与える反面それらの定め方の基準が与えられていないという欠点がある。テスト関数として用いた Rosenbrock, Beale, Box, Powell, Wood と Enzyme の関数に対して我々が定めた値は Table 2.1 ~ 2.3 に示されているが、ある程度の試行錯誤的な決定に頼らざるを得ないという難しさが残されている。我々の使用経験から言えば、刻み幅の決定はそれに対応する変数の目的関数に対する感度 (sensitivity) によりなされるべきであると思われる。即ち感度の高い変数に対する刻み幅は小さく定め、逆に低い変数に対しては多少大きめに定めることが望ましい。また調節係数 α はほぼ 2.0 ~ 3.0 程度、 β は 0.5 が妥当と思われる。いずれにしてもこれらの量の決定法は今後の課題であろう。

制約条件を持つ問題に対する ROTAX の精度は 3 つのテスト問題でみる限り満足のいく解が得られており、Allran と Johnsen 法を他の最適化手法へ適用する 1 つの保証が得られた。重み W_i は境界面 $f_i(x) = 0$ 上での目的関数の値により決定されるべきもので、境界面上での関数値が小さい場合には大きく定める必要がある。

謝 辞

本プログラムの開発にあたり終始貴重な御意見と手法上の検討をいただいた篠原慶邦主任研究員をはじめとするシステム解析手法研究グループの皆さんに心から感謝の意を表します。

参 考 文 献

- (1) Rosenbrock H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", Computer J., 3, 175 (1960)
- (2) Hooke R. & Jeeves T.A., "Direct Search Solution of Numerical & Statistical Problems", J. Assoc. Comput. Math., 8, 212 (1961)
- (3) Box M.J., "A New Method of Constrained Optimization and a Comparison with other Methods", Computer J., 8, 42 (1965)

- (4) Cheney E.W. & Goldstein A.A., "Newton's Method for Convex Programming and Tchebycheff Approximation", Numer. Math., 1, 253 (1959)
- (5) Griffith R.E. & Stewart R.A., "Nonlinear Programming Technique for the Optimization of Continuous Processing Systems", Management Science, 7 (1961)
- (6) Allran R.R. & Johnsen S.E.J., "An Algorithm for Solving Nonlinear Programming Problems Subject to Nonlinear Inequality Constraints", Computer J., 13, 171 (1970)
- (7) Beale E.M.L., "On an Iterative Method for Finding a Local Minimum of a Function of More Than One Variable", Statistical Techniques Research Group, Princeton University, Tech. Report 25 (1958)
- (8) Box M.J., "A Comparison of Several Current Optimization Methods and the Use of Transformations in Constrained Problems", Computer J., 9, 67 (1966)
- (9) Hillstrom K.E., "A Simulation Test Approach to the Evaluation and Comparison of Unconstrained Nonlinear Optimization Algorithms", ANL-76-20 (1976)
- (10) Kowalik J. & Osborne M.R., "Methods for Unconstrained Optimization Problems", American Elsevier Publishing Company, Inc. (1968)
- (11) Wilde D.J. & Beightler C.S., "Foundations of Optimization", Prentice-Hall, Inc., Englewood Cliffs, N.J. (1967)
- (12) 鈴木忠和, "少変数の非線形最適化問題に対する一手法", JAERI-M7229 (1977)

Appendix 1

ROTAX Fortran Program List

```

C ROTAX, PROGRAM FOR SOLVING NONLINEAR OPTIMIZATION PROBLEM WITH NONLINEAR
C CONSTRAINTS BY AXES ROTATION METHOD. INEQUALITY CONSTRAINTS ARE
C TREATED AS THE PENALTY TERM TO MAKE NEW OBJECTIVE FUNCTION WHICH IS
C SUGGESTED BY R.R. ALLRAN AND S.E.J. JOHNSEN.
C
C INPUT QUANTITIES
C IDENT ... PROBLEM IDENTIFICATION
C N ... NO. OF VARIABLES
C NRES ... ( = 0 ) NO CONSTRAINTS ARE SUPPOSED
C           ( GT 0 ) NRES CONSTRAINTS ARE SUPPOSED
C ALPHA,BETA ... STEP SIZE CONTROL COEFFICIENTS
C TAU, EPS ... CONVERGENCE CRITERION
C SDLMO(I) ... INITIAL STEP SIZE FOR EACH DIRECTION
C X(I) ... STARTING POINTS OF VARIABLES
C
C DIMENSION X(25), XN(25), S(25,25), SDLAM(25), SLAM(25), Q(25,25),
C           V(25,25), SBAR(25,25), SCLAR(25), V1(25), NZERO(25), SDLMO(25)
C           , IDENT(18)
C READ(5,940) IDENT
C WRITE(6,950) (IDENT(I), I=1,18)
C 940 FORMAT(18A4)
C 950 FORMAT(1H0,10X18A4)
C READ(5,1000) N, ALPHA, BETA, TAU, EPS, NRES
C READ(5,1100) (SDLMO(I), I=1,N)
C WRITE(6,2500)
C 2500 FORMAT(1H0,50H-----)
C WRITE(6,3000) N, ALPHA, BETA, TAU, EPS
C 3000 FORMAT(1H0,7X'N =', I3,12X'EXPANSION FACTOR',E15,5,5X'CONTRACTION F
C 1ACTOR',E15,5/1H0,5X'TAU =',E12,5,11X'EPS =',E12,5/1H0,10X'LAMDA')
C WRITE(6,3100) (SDLMO(I), I=1,N)
C 3100 FORMAT(1H ,8E15,5)
C BEGIN WITH THE NEW STARTING POINTS
C 1 CONTINUE
C READ(5,1100) (X(I), I=1,N)
C WRITE(6,3200) (X(I), I=1,N)
C 3200 FORMAT(1H0,'STARTING POINTS =',6E15,5)
C CALL CLOCKM(JT1)
C CHECK IF THE PROBLEM HAS ANY CONSTRAINTS OR NOT
C IF(NRES.LE.0) GO TO 3
C KX=0
C 2 KX=KX+1
C RK=-(KX**3)
C 3 CONTINUE
C INITIALIZATION
C DO 10 I=1,N
C DO 5 J=1,N
C 5 S(I,J)=0,
C 10 S(I,I)=1.0
C ITN=0
C ICOR=0
C 500 DMAX=0,
C SEARCH THE OPTIMUM POINT IN GIVEN ORTHOGONAL COORDINATE SYSTEM
C NTR=0
C DCOR=ICOR
C REDUCE THE STEP SIZE BY MULTIPLYING EXP(-DCOR)

```

```

DO 11 K=1,N
11 SDLAM(K)=SDLM0(K)*(EXP(-DCOR) )
   ICOR=ICOR+1
   DO 15 I=1,N
15 SLAM(I)=0.
C   BEGIN NEW ITERATION
400 CONTINUE
   NTR=NTR+1
   ITN=ITN+1
   NSUCS=0
   CALL MODEL(N,F,X,RK)
   DO 100 I=1,N
   DO 20 J=1,N
20  XN(J)=X(J)+SDLAM(I)*S(J,I)
   CALL MODEL(N,F1,XN,RK)
   IF(F1-F)25,25,30
C   SEARCHING PROCESS HAS SUCCEEDED FOR I-TH DIRECTION
25  F=F1
   NSUCS=NSUCS+1
   DO 26 J=1,N
26  X(J)=XN(J)
   SLAM(I)=SLAM(I)+SDLAM(I)
   SDLAM(I)=SDLAM(I)*ALPHA
   GO TO 50
C   SEARCHING PROCESS HAS FAILED
30  SDLAM(I)=-BETA*SDLAM(I)
50  IF(ABS(SDLAM(I)),GT,DMAX)DMAX=ABS(SDLAM(I))
100 CONTINUE
2200 FORMAT(1H ,5X I3,E20,6,5X6E15,5)
   IF(NSUCS,GT,0)GO TO 400
   IF(NTR,EQ,1) GO TO 400
C   CHECK IF THE PROBLEM HAS CONVERGED BY MAX(LAMDA(I)),LE,TAU
   IF(DMAX,LT,TAU)GO TO 900
   DO 110 I=1,N
   NZERO(I)=0
   IF(SLAM(I),EQ,0,)NZERO(I)=I
110 CONTINUE
   @NORM=0.
C   CHECK IF THERE ARE ANY DIRECTIONS WHERE SEARCH HAS FAILED IN A GIVEN
C   SYSTEM
   J1=0
   DO 125 J=1,N
   IF(NZERO(J),EQ,0) GO TO 125
   J1=J1+1
   DO 120 I=1,N
   @ (I,J1)=0.
   DO 120 K=J,N
120 @ (I,J1)=@(I,J1)+S(I,K)*SLAM(K)
125 CONTINUE
   JP(J1,E0,0)GO TO 900
   N1=J1
C   CHECK IF THE PROBLEM HAS CONVERGED BY NORM (@(I)) ,LE, EPS
   @NORM=0.
   DO 130 I=1,N
   V(I,1)=@(I,1)

```

```

130 QNORM=QNORM+Q(I,1)*Q(I,1)
SQRNM=SQRT(QNORM)
IF(SQRNM.LT.EPS)GO TO 900
C GENERATE A NEW ORTHOGONAL COORDINATE SYSTEM FROM Q(I) BY
C GRAM - SCHMIDT PROCEDURE
DO 135 I=1,N
135 SBAR(I,1)=V(I,1)/SQRNM
IF(N1.LE.1) GO TO 155
DO 150 J=2,N1
J1=J-1
VNORM=0.
DO 140 K=1,J1
SCLAR(K)=0.
DO 140 I=1,N
140 SCLAR(K)=SCLAR(K)+Q(I,J)*SBAR(I,K)
DO 145 I=1,N
V1(I)=0.
DO 146 K=1,J1
146 V1(I)=V1(I)+SCLAR(K)*SBAR(I,K)
V(I,J)=Q(I,J)-V1(I)
145 VNORM=VNORM+V(I,J)*V(I,J)
VNORM=SQRT(VNORM)
DO 147 I=1,N
147 SBAR(I,J)=V(I,J)/VNORM
150 CONTINUE
155 J1=N1
DO 160 J=1,N1
DO 160 I=1,N
160 V(I,J)=SBAR(I,J)
DO 170 J=1,N
IF(NZERO(J).EQ.0) GO TO 170
J1=J1+1
DO 165 I=1,N
165 V(I,J1)=S(I,J)
170 CONTINUE
DO 180 J=1,N
DO 180 I=1,N
180 S(I,J)=V(I,J)
GO TO 500
900 CONTINUE
C OUTPUT ROUTINE
CALL CLOCKM(JT2)
JTIM=JT2-JT1
CALL MODEL(N,F,X,RK)
WRITE(6,2100)(X(I),I=1,N)
WRITE(6,2000)F ,ITN,ICOR,JTIM
WRITE(6,2500)
IF (KX.LE.NRES) GO TO 2
GO TO 1
1000 FORMAT(12.4E10.5,13)
1100 FORMAT(7F10.5)
2000 FORMAT(1H0, 'FUNCTION VALUE =',E15.6,5X'ITERATIONS NO. =',I4/1H0,
1'NCORE =',I4,5X,'EXEC. TIME =',I7,2X'(MILLI - SEC.))
2100 FORMAT(1H , 'OPTIMUM POINTS =',6E15,5)
STOP
END

```

Appendix 2

Example of User's Routine MODEL for Wilde's Problem

```
SUBROUTINE MODEL(N, F, X, R)
DIMENSION X(25)
A = -EXP( ( X(1) - 1 ) ** 2 + ( X(2) - 2 ) ** 2 )
B = EXP( R * X(1) ) + EXP( R * X(2) )
  + EXP( R * ( X(1) - X(2) * X(2) ) )
  + EXP( R * ( -EXP( -X(1) ) + X(2) ) )
  + EXP( R * ( -2 * ( X(1) - 1 ) ** 2 + X(2) ) )
F = A + 50 * B
RETURN
END
```