

Title	プログラム・ライブラリィの追加登録
Author(s)	
Citation	大阪大学大型計算機センターニュース. 1979, 33, p. 61-102
Version Type	VoR
URL	https://hdl.handle.net/11094/65414
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

プログラム・ライブラリィの追加登録

名古屋大学より譲渡されたものと、大阪大学教養部小谷恒之氏の作成による以下のプログラムが、センター・ライブラリィとして6月1日より使用可能となります。

C2	GJMNKS, GJMNKD	ガーサイド・ジャラット・マックの方法 による実係数代数方程式の解法	
СЗ	JACELS, JACELD	ヤコビ 楕円関数 sn, cn, dn	
C3	PN, DPN PNM, DPNM	ルジャンドル多項式 ルジャンドル陪多項式	
СЗ	CGAMMA, DCGAMA	複素変数のガンマ関数	名古屋大学より
C3	ACND, DACND } ACNDC, DACNDC }	累積正規分布関数とその余関数の逆関数	譲受
C3 C3	AERF, DAERF AERFC, DAERFC }	誤差関数及びその余関数の逆関数	
C4	BROYDS, BROYDD	Broyden の方法による非線形連立一次方 程式の解法	
F5	MINVS, MINVD	行列の逆転	
C7	SIMPLX	関数の極小化(SIMPLEX法)	<i>II</i> No.14
С7	DAFLEP	関数の極小化(DAVIDONの方法)	作成者 · 大阪大学教養部 小谷 恒之
C7	POWLS	関数の極小化(POWELL の最小二乗法)	

次に使用方法について説明します。引数の説明の中で、実数型(倍精度実数型)とあるのは、倍精度のライブラリィを使用するときには、その英字名を倍精度実数型のデータとして宣言しなければならないことを意味します。これらのランクはすべて2です。

分類コード	ガーサイド・ジャラット・マックの方法による実係数代数方程式の解法						
C 2	単精度	CALL	GJMNKS (A, N, B, C, ILL)				
	倍精度	CALL	GJMNKD(A, N, B, C, ILL)				

SOLUTION OF POLYNOMIAL EQUATIONS WITH REAL COEFFICIENTS BY GARSIDE—JARRAT—MACK METHOD

I. 目 的

複素係数の代数方程式の解法として定評のある Garside — Jarrat — Mack の方法を実係数の方程式の解法として、二宮・門脇が改良したものをサブルーチン化したものである。原方法のもつ頑丈さと複素計算の実数化によるスピード・アップが組み合わされており、また、3次、4次の方程式の解法としてカルダノ法、ブラウン法が取り入れられているので、汎用の代数方程式ルーチンとして推奨できる。

2. 使 用 法

A ·············入力。実数型 (倍精度実数型) 一次元配列名。A(K) , $K \ge N+1$ 。

代数方程式の係数を最高次の係数から順に入力する。保存されない。A(1) × 0。

N ·············入力。整数型変数名。代数方程式の次数。保存される。 N≥1。

B············出力。実数型(倍精度実数型)一次元配列名。 B(K), $K \geq N$ 。

代数方程式の根の実部が出力される。根は、 かかむね絶対値の小さいものから求められ、B(N), B(N-1) …と逆順に格納される。

 C 出力。実数型(倍精度実数型)一次元配列名。 $\mathrm{C}(\mathrm{K})$, $\mathrm{K} \geq \mathrm{N}$ 。

代数方程式の根の虚部が出力される。根の計算順序と格納の仕方はBに同じ。

ILL出力。整数型変数名。

正常に計算が終了したとき 0, N<1又はA(1)=0 のとき 30000。滅次されたK次方程式を処理中に 200 回反復しても未収束のとき、ILL=K となる。

3. 備 考

重根や近接根などに対しても、原理的には収束率が変わらないという原方法の良さを受けついでいて、ほとんどの実際的な方程式に対して、途中で収束不能となることなく、方程式の条件が許す限りの精度で根が計算される。この意味で非常に頑丈であると言える。また、原方法では、実係数の方程式に対しても複素計算を必要としたが、これを実計算で行えるように改良したため

計算量は半減し、スピードが倍加された。

3次、4次の低次方程式に対しては、それぞれの専門の方法であるカルダノ法及びブラウン法 (フェラリ法の変形)を取り入れているので、これらの方程式に対しても専門ルーチンに遜色が ない。このような意味で、本ルーチンはベァストー法に勝る汎用ルーチンであるということがで きる。

代数方程式は悪条件となることが多いので、よほど低次の場合を除いては、倍精度用ルーチン GJMNKD を用いるのが安全である。

参考文献

G. R. Garside, P. Jarrat and C. Mack: "A New Method for Solving Polynomial Equations," Computer Journal, Vol. 11 (1968)

作成者 名古屋大学工学部 二宮市三。

分類コード		+	コビ楕円関数	s n,	en,	d n				
	単精度	CALL	JACELS (U,	АK,	SN,	CN,	DN,	QP,	ILL)
C 3	倍精度	CALL	JACELD (U,	AK,	SN,	CN,	DN,	QP,	ILL)

JACOBIAN ELLIPTIC FUNCTIONS sn, cn, dn

I. 目 的

JACELS (JACELD) は、変数 u , 母数の自乗 $_k$ 2を単(倍)精度で与えて、ヤコピ楕円関数 $sn(u, k^2)$, $cn(u, k^2)$, $dn(u, k^2)$ 及び第1種完全楕円積分 $K(k^2)$ を単(倍)精度で計算するサブルーチン副プログラムである。

2. 使 用 法

AK ……入力。実数型(倍精度実数型)変数名。 母数の自乗 k^2 の値を表す。保存される。

SN -----出力。実数型(倍精度実数型)変数名。 sn 関数の値が入る。

CN -----出力。実数型 (倍精度実数型) 変数名。 cn 関数の値が入る。

DN ·······出力。実数型 (倍精度実数型) 変数名。 dn 関数の値が入る。

QP ………出力。実数型 (倍精度実数型) 変数名。 k の値が入る。ただし AK = 1 のときは 0 とする。

ILL …… 出力。整数型変数名。U,AK についての制限が満たされ、正常に計算が行われた とき 0 ,そうでないとき 1 となる。

3. 備 考

 $0 \le AK \le 1$, $\left| U \right| \le QP$ でなければならない。

ザルツァーの算術幾何平均法による。

AK には母数 kではなくて、 k^2 を入れることに注意されたい。

参考文献

H. E. Salzer: "Quick Calculation of Jacobian Elliptic Functions", CACM, Vol. 5, p. 399, (1962)

作成者 名古屋大学工学部 二宮市三

分類コード	ルジャンドル多項式及びルジャンドル陪多項式						
0.9	単精度 Y=PN(N, X) Y=DPN(N, X)						
C 3	倍精度 Y=PNM(N,M,X) Y=DPNM(N,M,X)						

LEGENDRE AND ADJOINT LEGENDRE POLYNOMIAL

I. 目 的

PN(DPN) は、整数nとxを単(倍)精度で与えて $P_n(x)$ を単(倍)精度で計算する関数副プログラムである。

PNM(DPNM) は、整数n, m及びx を単(倍)精度で与えて $P_n^m(x)$ を単(倍)精度で計算する関数副プログラム。ただし、 $P_n(x)$ はルジャンドル多項式、 $P_n^m(x)$ はルジャンドル陪多項式である。

2. 使 用 法

N, M……整数型変数名。

X………実数型(倍精度実数型)変数名。

英字名は、上記の式を参照。

PN, DPN, PNM, DPNM は関数副プログラムである。したがって、DPN, DPNMは倍精度実数型として宣言しなければならない。

3. 備 考

計 算 法

(1) PN(DPN)

- (1) n < 0 ならばエラーとする。
- (2) n = 0 \$ 5 | $t^{p}_{n}(x) = 1$ \$ 2 \$ 5 \$ 6.
- (4) $n \ge 2$ ならば $P_0(x) = 1$, $P_1(x) = x$ より出発して、漸化式

$$P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x)$$

を2~nなる kについて順次適用する。

(2) PNM(DPNM)

- (2) m=0 ならば $P_n^m(x) = P_n(x)$ と計算する。
- (3) m>0 ならば $P_n^m(x)=(1-x^2)^{\frac{m}{2}}$ $P_n^m(x)$ と計算する。
- (4) $n = m \pi \text{ bid } P_n^m(x) = 1, 3, \dots, (2n-1) \text{ bids}.$
- (5) n=m+1 & Sid $P_n^m(x)=1$, 3, ..., (2n-1) $x \in \mathcal{F}_0$
- (6) $n \ge m+2$ ならば $P_m^m(x)=1$, 3, …, (2m-1), $P_{m+1}^m(x)=1$, 3, …, (2m+1)x より出発し、漸化式

$$P_k^m(x) = \frac{2 \cancel{k} - 1}{\cancel{k} - m} x P_{k-1}^m(x) - \frac{\cancel{k} + m - 1}{\cancel{k} - m} P_{k-2}^m(x)$$

 $\epsilon m + 2 \sim n$ なる ℓ について順次適用する。

倍精度の関数を使用するときは、そのプログラム単位で、その関数名を倍精度実数型として宣言しなければならない。

本ルーチンのルジャンドル陪多項式の定義は

$$P_n^m(x) = (1-x^2)^{\frac{m}{2}} \frac{d^m P(x)}{dx^m}$$

である。この定義と異なる定義もあるので注意されたい。

作成者 名古屋大学工学部 二宮市三 名古屋大学大型計算機センター 秦野寛世

分類コード	複素変数の ガンマ関数
С 3	単精度 Y=CGAMMA(Z)
િક	倍精度 Y=DCGAMA(Z)

GAMMA FUNCTION FOR COMPLEX ARGUMENTS

I. 目 的

CGAMMA (DCGAMA) は \mathcal{S} を (倍精度) 複素数 として与えて、 $\Gamma(\mathcal{S})$ を (倍精度) 複素数 として計算する関数副 プログラムである。

2. 使 用 法

CGAMMA, DCGAMAは、関数副プログラムである。したがって、CGAMMA は 複素数型の、DCGAMA は倍精度複素数型の宣言を必要とする。

3. 備 考

引数の値は、0又は負の整数でないこと。

引数が制限を破った場合はエラーとし、エラー・メッセージを印刷する。更に、関数値を強制的に0とし、計算を続行する。

精 度 CGAMMA は約7桁、DCGAMA は約17桁。

計 算 法

 $R_e(z)$ <0 ならば、反転公式

$$\Gamma(z) = \pi / (\Gamma(1-z)\sin \pi z)$$

により、 $R_e(z) \ge 0$ の場合に還元する。

 $R_e(z) \ge 0$ の場合

- (a) |z-1| < 1 ならば $1/\Gamma(z)$ を(z-1)のべき級数で計算する。
- (b) $|z| \ge 4$ ($|z| \ge 8$) ならば漸近級数 $\log \Gamma(z) = (z-1/2)\log z + \log \sqrt{2\pi} z + \sum_{n=1}^{\infty} \operatorname{Cn}/z^{2n-1}$ を利用する。
- (c) (a), (b)のいずれでもない場合は、漸化式 $\Gamma(z)=\Gamma(z+1)/z$ を必要なだけ反復して、 (b)の場合に還元する。

作成者 名古屋大学工学部 二宮市三

分類 コード	累積正規分布関数とその余関数の逆関数							
_	単精度	Y = A C N D (X)	Y = A C N D C (X)					
C 3	倍精度	Y = D A C N D (X)	Y = D A C N D C (X)					

THE INVERSE OF THE CUMULATIVE NORMAL DISTRIBUTION FUNCTION AND ITS COMPLEMENT

I. 目 的

ACND(DACND) $\}$ は、x を単(倍)精度で与えて、 $\{ \phi^{-1}(x) \}$ を単(倍)精度で計算する関数副プログラム。 ただし、 $\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$ $\psi(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-t^2/2} dt$, $\phi(x) + \psi(x) = 1$

であり、 $\phi^{-1}(x)$, $\psi^{-1}(x)$ は、それぞれ、 $\phi(x)$, $\psi(x)$ の逆関数である。

2. 使用法

X ············ 実数型(倍精度実数型)変数名。

ACND, ACNDC, DACND, DACNDC は関数副プログラムである。したがって DACND, DACNDCは倍精度実数型として宣言しなければならない。

3. 備 考

0 < X < 1 restricted reserves.

精 度 ACND 及び ACNDC は約7桁、DACND 及び DACNDC は約17桁。

引数が制限を破った場合にはエラーとし、エラー・メッセージを印刷する。更に、関数値を強制的に0として計算を続行する。

ACND(DACND)

 $\phi^{-1}(x) = -\sqrt{2} \text{ erfc}^{-1}(2x)$ を用いて、AERFC(DAERFC) に帰着せしめる。ACNDC(DACNDC)

 $\psi^{-1}(x)=\sqrt{2} \ {
m erfc}^{-1}(2x)$ を用いて、AERFC(DAERFC) に帰着せしめる。 作成者 名古屋大学工学部 二宮市三

分類コード		誤差関数及びその余関数の逆関数						
	単精度	Y = A E R F (X)						
0.0		Y = A E R F C (X)						
C 3	倍精度	Y = D A E R F (X)						
		Y = D A E R F C (X)						

THE INVERSE OF THE ERROR FUNCTION AND ITS COMPLEMENT

I. 目 的

AERF (DAERF) AERFC (DAERFC) は、x を単(倍)精度で与えて、 $\left\{ \begin{array}{l} \operatorname{erf}^{-1} \\ \operatorname{erfc}^{-1} \\ \end{array} \right\}$ を単(倍)精度で計算する関数副プログラム。ただし、 $\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^{2}} dt$ $\operatorname{erfc} x = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^{2}} dt$ $\operatorname{erf} x + \operatorname{erfc} x = 1$

であり、 $\operatorname{erf}^{-1}x$, $\operatorname{erf} c^{-1}x$ は、それぞれ、 $\operatorname{erf} x$, $\operatorname{erf} c x$ の逆関数である。

2. 使 用 法

X……実数型(倍精度実数型)変数名。

AERF, AERFC, DAERF, DAERFC は関数副プログラムである。したがって、 DAERF, DAERFCは倍精度実数型として宣言しなければならない。

3. 備 考

AERF(X), DAERF(X) に対しては、-1 < X < 1。

AERFC(X), DAERFC(X) に対しては、0 < X < 2。

引数が制限を破った場合にはエラーとし、エラー・メッセージを印刷する。更に、関数値を強制的に0とし、計算を続行する。

精 度 AERF 及び AER FC は約7桁、DAERF 及び DAERFC は約17桁。

計 算 法

- 1 AERF(DAERF)
 - (1) $|x| \ge 1$ $x \le 1$ $x \le 1$
 - (2) $|x| \leq 0.8$ ならば有理近似式により $\operatorname{erf}^{-1} x$ を計算する。
 - (3) |x| > 0.8 ならば $\operatorname{erf}^{-1} x = \operatorname{erfc}^{-1} (1-x)$ により AERFC(DAERFC) を引用する。
- ② AERFC(DAERFC)
 - (1) $x \le 0$ 又は $x \ge 2$ ならばエラーとする。
 - (2) $0.2 \le x \le 1.8$,すなわち、 $|1-x| \le 0.8$ ならば $\mathrm{erfc^{-1}}(1-x)$ により AERF (DAERF) を引用する。
 - (3) |1-x| > 0.8 ならば $\beta = \sqrt{-\log_2 x(2-x)}$ を計算し、 $\operatorname{erfc}^{-1}x = \operatorname{sign}(1-x) \cdot \beta \cdot \operatorname{R}(\beta)$ と計算する。
 - (4) $R(\beta)$ の計算は β の値に応じて三つの場合に分け、それぞれの有理近似式を用いて行う。

参考文献

A. J. S trecok: "On the Calculation of the Inverse of the Error Function", Math. Comp. Vol. 21,(1967)

作成者 名古屋大学工学部 二宮市三

分類コード		BROYDENの方法による非線形連立一次方程式の解法									
	単精度	CALL	BROYDS (X, N, H, KH, FN, LF, NF, EPS,								
C 4		FM, ILL)									
04	倍精度	C A L L	BROYDD(X, N, H, KH, FN, LF, NF, EPS,								
			FM, ILL)								

SOLUTION OF SYSTEMS OF NON-LINEAR EQUATIONS BY BROYDEN'S METHOD

I. 目 的

初期値が与えられたとき、非線形連立方程式 $f_1(x_1\ ,\ \cdots,\ x_n)=0$ ($i=1\ ,\ 2\ ,\cdots,\ n$) を B royden の 反復法で解くためのサブルーチン副プログラムである。

2. 使 用 法

- X ··············入出力。実数型(倍精度実数型)一次元配列名。X(K), K≥N。 初期ベクトルを入力すると解ベクトルが出力される。
- N ………入力。整数型変数名。方程式の元数。保存される。 $0 < N \le 100$ 。
- H·······作業領域。実数型(倍精度実数型)二次元配列名。 $H(K1, K2), K1 \ge N, K2 \ge N$ 。
- KH -----------入力。整数型変数名。H の配列宣言 に * かける第 1 添字の値。保存される。 KH \geq N。
- FN ………入力。サブルーチン手続名。位置ベクトルX が与えられたとき、N個の方程式の値から成るベクトルYを計算するための FN(X, Y) の形のサブルーチン。この引数に対する実引数は、本ルーチンを引用するプログラム単位で EXTERNAL宣言を必要とする。
- LF ………入力。整数型変数名。関数サブルーチン引用回数の上限。保存される。 $LF>N+1\, .$
- NF ……出力。整数型変数名。関数サブルーチン引用回数。
- EPS ………入力。実数型 (倍精度実数型)変数名。収束判定常数。保存される。EPS>0。
- FM ··········出力。実数型 (倍精度実数型) 変数名。方程式の残差 2 乗平均の平方根。
- ILL ……出力。整数型変数名。正常に計算が終了したとき 0 , NF>LF となっても収束しないとき 1 ,入力引数が条件を満足しないとき 30000 。

3. 備 考

非線形連立方程式は一般に多くの解を持つので、得られた解が果たして目的の解であるかどうかは検討の必要がある。目的の解に確実に収束させるためには、良い初期値が是非とも必要である。

最小値(又は最大値)が極値である場合の最小化問題は、勾配ベクトルに関して非線形連立方程式となり、逆に、非線形連立方程式 $f_i=0$ (i=1,...,n) は $\sum f_i^2$ を考えれば最小化問題となる。これを考慮して、それぞれの場合に適する計算法及びプログラムを選択すべきである。本ルーチンは、その中で逆行列ルーチンMINVS (MINVD)を引用している。

参考文献

C. G. Broyden: "A Class of Methods for Solving Nonlinear Simultaneous Equations", Math. Comp. Vol. 19, pp, 577-593 (1965)

作成者 名古屋大学工学部 二宮市三

分類コード		ŕ	亍 列	0	逆	転			
70.5	単精度	C A L L	MIN	VS(A,	К,	Ν,	EPS,	ILL)	
F 5	倍精度	CALL	MIN	V D (A,	Κ,	N,	EPS,	ILL)	

INVERSION OF MATRICES

I. 目 的

Gauss – Jordan の消去法によって、与えられた行列の位置に作る。この際、ビボット選択のために必要ならば、行の交換を行う。

2. 使 用 法

A……入出力。実数型(倍精度実数型)二次元配列名。

与えられた行列を入れると、本ルーチンにより計算された逆行列が入る。

K入力。整数型変数名。A の配列宣言における第1 添字の値。保存される。 $K \ge N$ 。

N ……入力。整数型変数名。A の次数。保存される。 $2 \le N \le 300$ 。

EPS ……入力。実数型(倍精度実数型)変数名。

行列の特異性の判定常数。ピボット要素の絶対値がこの常数より小さいとき特異であると判定して、計算を中断する。保存される。EPS>0。

ILL ……出力。整数型変数名。正常に計算が行われたとき 0, K, N, EPSに関する制限が 破られたとき 3000, 特異と判定されたときはビボット要素の番号。

3. 備 考

- (1) 行列の要素の絶対値の間に大きな懸隔があるときは、あらかじめ正規化することが精度を保つ上で望ましい。
- (2) EPS 標準的 α を値としては、行列の要素の絶対値の代表的な値を α とすると、MINVS (MINVD)に対しては α × 10^{-7} (α × 10^{-10}) が適当である。

作成者 名古屋大学工学部 二宮市三

分類コード	関数の極小化(SIMPLEX法)							
	単精度 CALL SIMPLX(FCN, N, X, EPSF, EPSX, ICONT,							
C 7	NCUT, JXD, XD, JOKEN, XMAX, XMIN,							
	IPRINT, LMOD, FMIN, KAISU, W, NW)							

MINIMIZATION OF FUNCTION BY THE SIMPLEX METHOD

I. 目 的

関数の極小値を求めるのに、シンプレックス法を用いる。関数値の変化とN個の変数の最大の変化率とが、それぞれ与えられた上限値より小さくなるまで、このプログラム内で自動的に反復される。

この方法の主目的は、関数の極小値を与える変数の組(X)を最終的に決定するというより、 関数の変化を放物線で近似できるような X の領域を広範囲にわたって探し、浅い " くぼみ " に落 ち込まないようにすることを主眼としている。しかも方法が単純で、関数の変化の様子を知る手 懸りを得るのに適しているので、他の方法(例えば POWLS またはDAFLEP など)より前の段 階で用いるのが適切である。

なお、XのN個の各成分の上下限をそれぞれ与えることもできるようになっている。

2. 入力データ

N……極小化を行う関数が含む変数の数。保存される。

X …………N 次元変数空間のベクトルX の成分。実数型1 次元配列で、大きさはX (K), $K \ge N$ 。

入力として、Xの初期値の成分 ($X(J)=X_J$, J=1, …N) を与える。 出力として、極小値に対応するX(J)が入っている。

EPSF によって、関数値そのものか、その変化かのいづれかの(使用者の選択 EPSX できめられる)上限が指定される。

EPSX は各X(J) の 精度を定める量で、それらの変化率の最大のものの上限を与える。いづれも実数値で保存される量。(7. 停止の条件参照。)

ICONT…… ICONT = 1 なら、関数の変化が極小値附近で火口型になるまで新しいシンプレックスを求める操作を続けることを要求する。(9. 使用上の注意(3)参照) ICONT ≒ 1 なら、このような条件が存在しないことを意味する。
ICONT は保存されず、出力として、

ICONT=1: 収束条件を満足して計算が終ったことを示す。

ICONT=-1:収束しないで終了したことを示す。

NCIIT ……このプログラム内で関数の値を計算する回数の上限を与え、停止の条件の一つに なる。実行上、(NCUT + 4N) 程度まで計算されることがある。整数値で保存 される。

JXD $\}$ 初めのシンプレックスを形成するため、XのJ成分を \pm 〔X D(J)*X(J)]だ XD け動かして、関数の極小値を探す。もしX(J)=0であれば、 $\pm XD(J)$ だけ動 かす。

JXD=1の場合(入力):

N個のXD(J)を入力として与えたことを示す。 なおプログラム内に

 $XD(J) \ge EPSX$ の制限が内蔵されている。このXD(J) は保存されないが、 必要なら、作業領域に $W(N+J)(J=1, \dots N)$ として記憶されている。

J X D ≒ 1 の場合(入力):

XDの値について指定しない場合で、プログラム内で自動的に

$$XD(J) = 0.1 (J = 1, \dots N)$$

の値をとる。

XDは実数型一次元配列で、大きさはXD(K), $K \ge N$ 。

出力としては、一番最後のシンプレックスの(N+1)個のXの成分X(J)の最 大値と最小値の差が入っている。

JOKEN \ ... JOKEN ≒ 0 の場合:

XMAX XMIN

X(J)のとりうる最大値 (XMAX)と最小値 (XMIN) とを入力として与えたこと を意味する。いづれも実数型1次元配列で、大きさはXMAX(K)もXMIN(K) も、K≥Nで保存される。

JOKEN = 0 の場合:

XIC制限をつけない場合で、 XMAX やXMIN の値は無視されるが、次元数だけ は確保する必要がある。

IPRINT、…出力データ や種々の途中経過の情報の書き出しを指定する整数値で保存される。 LMOD IPRINT = 0 (LMOD ≥ 1 で任意) が標準形で、10回目毎のシンプレックス

につき、最小の関数値(FMIN)と対応するXの値とが書き出される。

IPRINT = 1, 2(LMOD≥1で任意) に応じて、途中経過を書き出す情報量が 滅る。IPRINT≥4ではなにも書き出さない。

IPRINT = -1 または-2 では、始めのシンプレックスの形成の経過や、

[LMOD] 回毎のシンプレックスのFMIN や対応するX その他の情報量が増加する。計算結果が異常な場合の検査用である。

KAISU----- とのプログラムに入る前に関数を CALL した回数を入れる。との入力 KAISU にとのプログラム内で関数を CALL した回数が加算されて出力される。

計算すべき関数……

極小化する関数を定義するサブルーチンとして

SUBROUTINE FCN(N, X, F)

DIMENSION X(N)

この SUBROUTINE のなかに求める関数の定義式 F= の形式のものが必要である。

なお、このSIMPLX を CALLする文のあるプログラムの中で、 FCN に対し EXTERNAL の宣言を必要とする。 EXTERNAL 文中の FCN, CALL SIMPLX の最初の実引数の FCN と SUBROUTINE の FCN の 3 箇所の FCN は、同じ文字でさえあれば、使用者の好みの名前のものでよく、 FCN という文字にとだわることはない。 (12.使用例参照)

3. 出力データ

FMIN 関数の最終の極小値で実数型。

X………入力データで説明した。 FMIN を与える変数 X(J) である。

ICONT……入力データで説明したように、+1と-1の値で、成功と失敗を示す。

4. 作業領域

W……作業用として用いられる実数型 1次元配列で、大きさはW(K), $K \ge NW$ 。

NW……作業領域の大きさをきめる整数で、

 $NW \ge N^2 + 8N + 1$

5. 使用サブルーチンと禁止文字群

入力データに記したように、関数F(X)を定義するサブルーチンFCNは使用者が作成する。それ以外にサブルーチンKOSIMPとXLIMIT,名前付共通ブロックINATOKの三つの名前が使われているので、これらの四文字群は使用者のサブルーチン名、共通ブロック名として使用できない。

6. 計算法の概略

このシンプレックス法の基本的な考えは、NELDER と MEAD によるものである。(流れ図まで入ったわかり易い解説がKOWALIKとOSBORNEの本の邦訳にある。)

ことで用いるのは、ヨーロッパ連合原子核研究所の F. JAMES & M. ROOS によって作成された(1976 年)プログラムを土台にして、小谷が修正(以下で*印をつけた項目)したものである。

原著に対し、本プログラムでかなり異なっているのは次のような諸点である。

- (1)* 初めのシンプレックスを定めるのに、XのN成分のそれぞれの方向での極小値を探して、より大きいシンプレックスを作って、局所的な、〈ぼみ、を避けるようにしてある。
- (2) * 初めのシンプレックスの各関数値のうち最小のものを FMIN とし、他のN個の関数値がすべて、10* FMIN ょり大きい場合は、FMIN κ 対応する \mathbf{X}_L を入力 \mathbf{X} とみなして再出発する。
- (3)*シンプレックスの形が小さくなったり、拡大しすぎるなど極端にひずみが生ずると、再出発して、改めてシンプレックスを求め直している。その際に、J番目の方向でより小さい関数値が求まると、それに対応するXを出発点とし、(J+1)番目の方向でより小さい値を求めて、出発のシンプレックスを決定している。
- $(4)^*$ シンプレックスの各点 X_k (k=1, …, N+1) のうち、関数値の最大値(FMAX) に対応する X_H を除外し、残りの X_k につき平均値を求める際、各点での関数値の逆数を重みとして、加重平均値を採用し、それを \overline{X} とした。
- (5) X_H と \overline{X} を結ぶ線上で、 $\overline{X}-X_H$ の 6 倍まで、より小さい関数値を探索している。
- (6)*シンプレックスを縮小する場合には、関数値の二番目に大きい値との関連を考慮して、関数を計算する回数を一回でも少なくなるようにした。
- (7) \mathbf{X}_H と $\overline{\mathbf{X}}$ との中点での関数値が FMAX より大きい場合には、関数の最小値を与えた \mathbf{X}_L を 出発点にとって、新しくシンプレックスを求め直して再出発している。
- (8)* 各成分 X(J)の最大値と最小値を指定できるようにした。

7. 停止の条件

(7-I) 収束して終了(成功,ICONT=1)

- (1) 関数値 $F(X) \le EPSF$ なら次の(3)の関門へ移る。ただし $F(X) \le 0.01*$ EPSF ならただ ちに停止する。
- (2) あるシンプレックスでの関数の最小値を FMIN (= YSTART) とし、その次に作られた新 しいシンプレックスでの FMIN が

 $0 < YSTART - FMIN \le EPSF$

の場合、次の(3)の関門へ移る。

(3) あるシンプレックスで、 FMIN を与える \mathbf{X}_L の \mathbf{J} 番目の成分を \mathbf{X}_L (\mathbf{J}) とする。シンプレックスを構成する ($\mathbf{N}+\mathbf{1}$) 個の \mathbf{X} の \mathbf{J} 番目の成分の最大値と最小値の差 (\mathbf{X} D(\mathbf{J})として出力される) を求め、

$$\mathrm{EPSX} > \!\! \mathrm{max} \, \big(\frac{\mathrm{XD} \, (\mathrm{J})}{\mathrm{X}_L \, (\mathrm{J})} \big)$$

であれば、(4)へ移る。この条件が10回満たされなければ、 X_L を入力Xとして再出発する。

(4) ICONT 1 なら、この関門は無条件通過で、成功として終る。

ICONT = 1 では、入力で説明したように、放物線近似が使えるような場合に終了。そうでなければ、再出発して継続される。

(7-2) 収束せずに終了(不成功, ICONT=-1)

- (1) 4回再出発して、上記の諸条件を満たさない場合。
- (2) 入力した KAISU(≡NPFN)を差しひいた KAISU, すなわち

 $KAISU - NPFN \ge NCUT$

で、停止する。

8. メッセージ (標準の IPRINT = 0 を主として述べる)

関数値 F(X) の始めと終りの値は FMIN として $IPRINT \le 3$ なら書き出される。また、収束条件として与えられた EPSF, EPSX,や NCUT 値は $IPRINT \le 0$ の場合に書き出される。その他で読んで自明と思われるものの説明は省略する。以下のメッセージの終りに括弧内に記した数字は IPRINT がその数字以下なら書き出されるメッセージである。(以下 $IP \equiv IPRINT$ と略)、 $IPRINT \ge 4$ ではメッセージは一切でてこない。

NZ 回目の再出発で、(8001+NZ)の数字でも示される。すぐ上の行に再出発の理由が書き出される。NZ=4までで停止の条件を満たさなければ、収束せずに終了する。

O FMIN IN THE PREVIOUS RUN WAS NOT CHANED. THIS TIME,

STARTING STEP = $Z (IP \le 0)$

再出発で最初のシンプレックスを決定する場合、前回に最小の関数値 (FMIN) が全然変化 してなければ、Xの J番目の成分を

$$X (J) \pm X D (J) * X (J) * Z$$

のように、Z倍だけ拡大したシンプレックスを求めたことを示す。

9. 使用上の注意

(1) IPRINT = 0 でまず最初に関数値の変化の概略を知るのがよさそうである。異常があれば、IPRINT = -1, LMOD = 10 位で、X の各成分についての依存性を調べて、入力X の値が適当か不適当かを判断することができる。

最終結果は入力Xにかなり強く依存するので、入力X をいろいろと変化させてみることが必要である。

- (2) との副プログラム SIMPLX をCALLする前に、関数値を求めておく必要はない。
- (3) ICONT = 1 で 入力した場合に、関数の変化を放物線で近似してよいかどうかを、X の名成分毎に知るには、作業領域で用いたW(K) から判別できる。すなわち、X(J) に関して、可ならW(J) = 1.0,否ならW(J) = -1.0 となっている(J = 1 , ……, N)。 この判定は、X の一つの成分のみを一割変化させて求めたF(X) の値が、FMIN の両側で共にFMIN より大きくなった場合に可としてある。
- (4) Xの各成分について、関数の依存性のもう少し精細な情報を得たい場合は、付録に述べる副 プログラムXERROR を CALL すればよい。
- (5) この方法は X_f の大きさにはよらないが、特に JXD=0 の場合、 X_f の一割の単位 (XD_f) で関数値の大小を比較しているので、N個の X_f のなかに極端に大きい数や小さい数が混在すると探索の領域がひずんでしまう。それを避けるため、 X_f の大きさをスケーリングさせて、大体の大きさを揃えてある方が望ましい。または $JXD \succeq 0$ として、各 X_f ごとに XD_f を与えるのがよい。

IO. 備 考

日本国内で利用できるシンプレックス法を用いたプログラムとしては、日立のサービス・ライブラリーの中にSIMPLXとして入っているものがある。

II. 参 考 文 献

- (1) J. A. Nelder and R. Nead, A simplex method for function minimization, Computer Journal 7(1965), 308.
- (2) J. Kowalik and M. R. Osborne, Methods for Unconstrained
 Optimization Problems (American Elsevier, New York, 1968).

 (邦訳:山本·小山「非線形最適化問題」(培風館, 昭和45年))
- (3) 小谷恒之,「非線型多変数関数の極小化」 大阪大学大型計算機センター・ニュース № 3 2 (1979)pp. 27.

12. 使用例

Rosenbrock の関数

```
F = \{ 10 (y-x^2) \}^2 + (1-x)^2
```

を例として示しておく $^{(3)}$ これは、x とy の値を与えて、F の極小値に対応するx=y=1 を求める問題である。* 印のものは付録のXERROR を用いる場合に必要なものである。

```
EXAMPLE ROSENBROCK'S FUNCTION
C
      DIMENSION X(2) a XD(2) a XMAX(2) a XMIN(2) a W(30)
      EXTERNAL FON
      N = 2
      READ(5,1010) (X(I),I=1,N)
 1010 FORMAT (2F10.5)
      EPSF = 1,0E-7
      EPSX = 2.0E=4
      ICONT = 0
      NCUT = 1000
      JXD = 0
      JOKEN=0
      IPRINT = 0
      NMOD = 5
      KAISU = 0
      NW = 30
      CALL SIMPLX(FCN, N, X, EPSF, EPSX, ICONY, NCUT, JXD, XD, JOKEN, XMAX,
     & XMIN, IPRINT, NMOD, FMIN, KAISU, W, NW)
      VARF = 0.03
      CALL XERROR (FCN, FMIN, N, M, X, W, NW, VARF)
      STOP
      END
```

SUBROUTINE FCN(N, X, F)
DIMENSION X(N)
FCA =10.0÷(X(2)÷X(1)÷X(1))
FCB =1.0÷X(1)
F = FCA*FCA + FCB*FCB
RETURN
END

付 録:変数 X の誤差を求めるプログラム

I. サブルーチンXERROR

関数の局所的な性質をXの各成分毎に調べられるように、SIMPLXとは別のCALL文で引き出すサブルーチンを別に用意してある。

CALL XERROR (FCN, FMIN, N, M, X, W, NW, VARF)

とのサブルーチンXERROR をCALLするプログラムでFCNをEXTERNAL 宣言しておくと
とが必要である。

2. 入力データ

FCN, FMIN, N, XはSIMPLXで説明したものと同じで、いづれも保存される。

計算すべき関数値もSIMPLX と同じく、SUBROUTINE FCN(N, X, F) で求めるので、 使用者が用意する必要がある。このサブルーチン名 FCN に関する注意も同様である。

Mは上記のFCNで求めたFが正しい関数値の 1M であることを示す整数値で、例えば、実験 データの χ^2 -f i t を求める場合であれば、データ数をM とし、正しい関数値の 1M をF とすれば見透しがよい。

VARFは、入力Xに対応するFMINから、標準値を

XSTAND = (1 + VARF) * FMIN

として決定し、この標準値を与えるように、ある成分の値を $X_K+\operatorname{ER}_\pm(\kappa)$,その他のX(J) ($J \succeq K$) は固定して, $\operatorname{ER}_+(K)$ を求めるためのもので、保存される。(第1図参照)

3. 出力データ

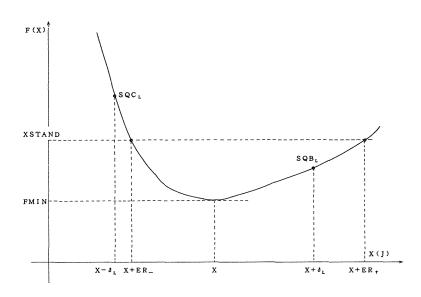
すべてメッセージとして書き出される。

4. 作業領域

Wは作業用としての実数型 1次元配列で、大きさはW(K), $K \ge NW$ 。 NW は作用領域をきめる整数で、 $NW \ge 6*N$ 。

5. 使用サブルーチン

関数値を与えるサブルーチンFCNはSIMPLXと同じものである。



第1図 関数の変化(Xの各成分が横軸)

$$X S T A N D = FM I N * (1 + V A R F)$$

 $\delta_L = (0.5)^{L-1} * 0.01 * |X|$
 $S QB_L = F(X + \delta_L)$
 $S QC_L = F(X - \delta_L)$

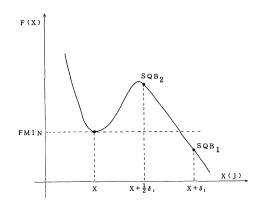
6. 計算法の概略

第 1 図に示されるように、 \mathbf{X} の \mathbf{J} 番目の成分に Δ_J を加減した関数値 \mathbf{F} ($\mathbf{X}\pm\Delta_J$) を求める。 $\mathsf{C}\mathsf{C}\tau\Delta_J = |\mathbf{X}_J| * 0.0 \ 1 \ (\mathbf{X}_J = \mathbf{0} \ \text{ の場合には} \ \Delta_J = \mathbf{0.0} \ 1 \)$ ととってある。 これらの関数値をそれぞれ $\mathbf{S}\mathbf{Q}\mathbf{B}_L$, $\mathbf{S}\mathbf{Q}\mathbf{C}_L$ とし、両者が \mathbf{FMIN} より大きければ、これら三点より放物

線近似により、 $\operatorname{ER}_{\pm}(\operatorname{J})$ を求める。 との場合には、メッセージに REPEAT $=\operatorname{L}$ と書き出される。添字 L は後述のくりかえしの回数 ($\operatorname{L} \leq 8$) である。

 SQB_L または SQC_L がXSTAND より小さい場合にはREPEAT=20+Lとなる。

添字Lは、例えばFMIN>SQB_L の場合には、第2図のような場合が考え



第2図 $FMIN > SQB_1$ の場合

られるので、

$$SQB_L = F(X + (0.5)^{L-1} * \Delta_I)$$

として、 $\operatorname{SQB}_L > \operatorname{FMIN}$ になるまでくり返す。このくり返しの回数が添字 L である。 SQC_L についても、

$$SQC_L = F(X - (0.5)^{L-1} * \Delta_I)$$

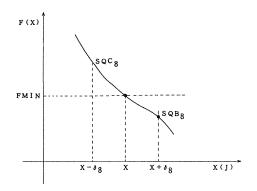
として、 $SQC_L > FMIN$ になるまでくり返す。

さらに、 SQB_L と SQC_L の値が、 FMINからの変化の%として表される。

$$SQ_{+} = \left(\frac{SQB_{L}}{FMIN} - 1 \right) * 100$$

$$SQ_{-} = (\frac{SQC_{L}}{FMIN} - 1) * 100$$

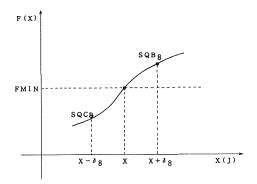
FMIN が火口型の底に対応せずに、第 $3 \sim 4$ 図のような、まだ斜面にある場合、それぞれ REPEAT = 30 + L または 40 + L となる。

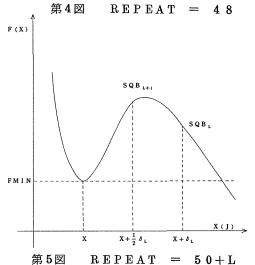


第3回 REPEAT = 38

一方、 $SQB_L>FMIN$ であっても、第 5図のような異常な場合も考えられるので、添字Lの場合、必ずL+1 を求めて、第 5図のようになっていないことが確認されている。 L=8 になっても、第 5図のようであれば、REPEAT=58 となる。同様なことが、 SQC_L でおこれば、REPEAT=68 と書き出される。

これら、REPEAT>30 では誤差





ERに意味がないので、

$$ER_{+} = SQB_{1}$$

$$ER_{-} = SQC_{1}$$

の値が入っている。

7. 停止の条件

入力データとして停止の条件を与えるものはないが、上述のXの各成分につき、L=8で終了する。

8. メッセージ

D. F. (=M-N)=-----(M-N)の値 J=1からNに対して、X, ER+, ER-, REPEAT, SQ+, SQ-の順に表にして書き出される。

9. 使用上の注意

- (1) REPEAT \geq 30 の場合は、関数の変化になんらかの異常があることを示している。これらの様子を精細にみるには、SIMPLEX法で、IPRINT \leq -1, JXD=1, NCUT \geq 10*N, LMOD=1と指定して、XD(J)に適当な小さい値(例えば 0.001)を与えて計算させればよい。
- (2) このXERROR をCALLする前には必ずXに対応する関数値 FMIN(≒0)が与えられていなければならない。

作成者 : 大阪大学教養部 小谷 恒之

分類コード		関数の極小化(DAVIDONの方法)							
C 7	単精度	CALL	DAFLEP (FCNG, N, X, EPSF, EPSX,						
			VTEST, NCUT, XD, JOKEN, XMAX, XMIN,						
			IPRINT, IGRAD, FMIN, KAISU, ICOVAR,						
			W, NW)						

MINIMIZATION OF FUNCTION BY THE METHOD OF DAVIDON—FLETCHER—POWELL—FLETCHER

I. 目 的

非線形の関数を極小にするN個の変数の組を求めるのに、DAVIDONによって提案され、 FLETCHERにより改良された方法を用いる。偏微分係数の値を利用しているが、数値的にそれ を計算するプログラムが含まれているので、使用者が偏微分係数を式で与えることは必ずしも必 要ではない。

関数値の変化と各変数の変化率の最大値とが、それぞれ与えられた上限値より小さくなるまで、 このプログラム内で自動的に反復される。

N個の変数のそれぞれにつき上下限を与えることもできるようになっている。

2. 入力データ

(2-I) SIMPLXと異なるもの。

VTEST ……共分散行列(V)の対角要素の平均変化率の上限を与える実数値で保存される (7. 停止の条件参照)。 普通 0.01 位の値をとっておけばよい。

XD最初に 2 階偏微分係数まで求めるため、ベクトルX で表したN 個の変数の成分 $X_{J} \ (\ J=1\ ,\ \cdots,\ N\)$ の変化率を与える。すなわち、J 番目の X_{J} の変化を Δ_{J} とし、関数 F (X)の 1 階偏微分係数を

$$\frac{\partial F}{\partial X_{I}} = \frac{F(X + \Delta_{I}) - F(X - \Delta_{I})}{2 \Delta_{I}}$$

で計算している。 ここで 1,は

$$\Delta_I = X_I * XD(J)$$

 $b \cup X_I = 0$ $restriction of the state of <math>\Delta_I = X D(J) \ge 2 + 2 + 2 = 3 + 2$

このX Dは正の実数型 1 次元配列で、大きさは X D(K), K≥N。保存されな

い。出力としては、最終の直線探索での X_J の変化がそれぞれ入っている。特別の事情がなければ、XD(J)=1 0 $^{-3}$ でよい。XD=1 0 $^{-4}$ (0.1)より小さい(大きい)と自動的に 1 0 $^{-4}$ (0.1) までとされる。

IGRAD ········IGRAD = 1 の場合:

途中での1階偏微分係数を、次に述べるサブルーチン FCNG 内で、使用者が 数値として与えていることを示す。

IGRAD ≒1の場合:

1 階偏徴分係数を、このプログラムに内蔵されているサブルーチン GRAD内で、直線探索での X_J の変化の $\frac{1}{10}$ か $|X_J|*10^{-4}$ かの小さい方 $(|X_J|*10^{-5}$ まで)を Δ_J として計算する。

ICOVAR ······ ICOVAR = 1 の場合(入力):

関数値が \mathbf{EPSF} (後述) $\mathbf{LSPMSEPSF}$ (後述) $\mathbf{LSPMSEPSF}$ の値と無関係に計算は停止される。

ICOVAR≒1の場合(入力):

SIMPLEX の場合と同じく、EPSFやEPSX 等の停止の条件に従う。

(7. 停止の条件参照)

との値は保存されず、出力の内容は3. 出力データで述べる。

NW……作業領域の次元数。(4. 作業領域参照)

計算すべき関数

……F(X)を定義するサブルーチンとして、使用者が用意する。

SUBROUTINE FCNG(N, X, F, G, L)

DIMENSION X(N), G(N)

が必要である。ここでLはプログラム内で指定されてくる数で、L=1または 0のいづれかである。

 $IGRAD \succeq 1$ なら、プログラム内でL=0 しか現れず、求める関数の定義式 (F=…)のみ与えればよい。微分係数G(K)(K=1, ..., N)=0.0は 必要としない。

IGRAD = 1 なら、L=0とL=1と両方がでてくる。L=1の場合に、1 階偏微分係数がG(K)(K=1,…N)で与えるように使用者側で用意されたい。L=0の場合、Fの値を与えることは IGRAD ≒ 1と同じである。 このDAFLEPをCALLするプログラムでFCNGに対しEXTERNAL 宣言文が必要である。EXTERNAL文、CALL DAFLEP文、SUBROUTINE FCNGの3箇所にでてくるFCNGは、使用者に都合のよい文字で揃えてあれば よく、FCNGにとだわらなくてよい。(12.使用例参照)

(2-2) SIMPLX と同じもの

 $\frac{N}{X}$ } …… SIMPLX に同じ。

EPSF } …… SIMPLX と同じ内容だが、数値は小さくしてよい。(9. 使用上の注意参照)

NCUT …… SIMPLX と同じ。実行上、(NCUT + 2N) 回程度まで関数値を計算すると とがある。

JOKEN XMAX XMIN SIMPLXと同じ。

IPRINT \cdots IPRINT = 0 が標準形で、IPRINT \ge 4 ではなにも書き出さないなど SIMPLX と同じ内容だが、LMODにあたるものはなく、直線探索の始まりで の FMIN, X_J とそれまでのKAISUとが、IPRINT \le 0 なら、書き出される。

KAISU SIMPLX と同じ。ただ、IGRAD = 1 の場合、偏微分を解析的な式で計算すれば、CPU時間は短縮されるが、回数としては 2 N回が加算される。

3. 出力データ

FMIN …… 関数 F(X)の極小値で実数型。

X FMINを与える変数の値。

ICOVAR ······ 1:条件を満たす解が得られた。

0 以下:不成功。

4. 作業領域

W……作業用として用いられる実数型一次元配列で、大きさはW(K), $K \ge NW$ 。

NW……作業領域の大きさをきめる整数で

$NW \ge N * (2 * N + 8)_{o}$

ととる。SIMPLXより少し大きいので注意せよ。

5. 使用サブルーチンと禁止文字群

入力データで記したように、関数 F(X)の定義式を与える $\left[IGRAD=1\right.$ の場合には微分係数 G(K)の定義式をも加えたものである $\right]$ サブルーチン FCNG は使用者が作成するが、それ以外にサブルーチンMATCOR、GRAD、INVERT の三種が内蔵されている。また名前付共通ブロック INATOL があるので、これら五組の文字の組合せは使用者のサブルーン名、共通ブロック名 として使用できない。

6. 計算法の概略

この方法は最初 DAVI DON (1) により提案され、明確な形にしたのは、FLETCHER & POWELL (2) である。1970年にFLETCHER (3) により、共分散行列(V)の近似を高めるのに、場合に応じて 二種類の式を使いわける方法(SWITCHING法)が提案されたので、それもとりいれた。 (4) 本プログラムはヨーロッパ連合原子核研究所の F. JAMES & M. ROOS によって1976年に 作成されたものを、小谷が更に修正したものである。(以下で*印を付した項目が小谷の修正である。)本プログラムでなされた特別の工夫は下記のような諸点である。

- (1) 共分散行列(V)につき、最初のステップで、単位行列を用いる代りに、対角要素に2階微分の逆数が用いられる。 V を正値行列にするため、2階微分係数が零または負になるような変数に関しては、関数が小さくなる方向に変数を変化させて正になるところを初期値として選び直す。ある範囲で探し出せない場合はV(I, I) = 1.0 とおく。
- (2) 直線探索を行なう方向は、J成分の1階微分をGS(J)として、

$$XD(I) = -\sum_{I=1}^{N} V(I, J) *GS(J)$$

と選ばれる。したがって、変数のスケールの選び方による不定性は入ってこない。

- (3) * 関数値を極小にする直線探索の精度は、 X_J の変化率が5%位から初め、極値に近づくにつれて精度があがるようにした。
- (4)* X_J を変化させて、その値が指定された上下限($XMAX_J$ や $XMIN_J$)の外へ出た場合には、 X_J =一定の超曲面上で、 X_J 以外の変数への依存性が調べられる。(この状態で 1 の回以上直線探索がくり返されると再出発させる。) X_J の値が上下限間に再びもどれば、 X_J も変化させるのはもちろんだが、二度目に上下限外へ出ると直ちに再出発させる。
- (5) Vが正値行列でなくなるような場合が発生すると、始めにもどって再出発する。
- (6)*関数Fの変化が EPSF より小さくなると、Xの変化率の最大値が EPSX より小さいかど

うかが検査される。この関門を5回までで合格しない場合は、始めから再出発する。

(7)*再出発に際して、最初の微分を求めるXDは再出発ごとに2倍、3倍と大きくされる。

7. 停止の条件

(7-I) 収束して終了(成功,ICOVAR=1)

- (1) 関数値 F(X) が EPSF以下になると、入力で ICOVAR=1 の場合は停止、 ICOVAR = 1 の場合は 0.01*EPSF なら停止し、それ以外は関門(3)へ移る。
- (2) 直線探索を始める前の関数値と変数の値をそれぞれFS, XXS(J)とし、探索後の対応 する値を FMIN, X(J)とする。さらにこの FMINとF(X)の極小値の推定値までの 距離をσ(ESTIMATED DISTANCE TO MINIMUM)とすると

$$FS - FMIN + \sigma < 0.1 * EPSF$$

ならば(3)へ移る。(文献(4)の式(4.12)参照)

直線探索をN回やった後であれば、

で(3)の関門へ移る。

(3) (1)か(2)を通過後、次の条件が満たされた場合に停止する。

$$\max_{1 \le J \le N} |1 - \frac{XXS(J)}{X(J)}| \le EPSX$$

(4) 共分散行列の対角要素 V (J, J) の値を、直線探索前後でそれぞれ V B (J), V A (J) として、次の条件が成立する場合 V に使止する:

TRACE =
$$\left(\frac{1}{N}\sum_{i=1}^{N} \left\{ \frac{VA(J) - VB(J)}{VA(J) + VB(J)} \right\}^{2} \right)^{1/2} < VTEST$$

- (5) 極小値への推定距離 σ が0.01*EPSFより小さい場合に停止する。
- (6) 3回再出発したが、直線探索で X_J を変化させても、F(X)の値が不変である場合に停止する。途中の極小値で停滞している可能性がある。

(7-2) 不成功で終了。(不成功の原因はIER=として, $IPRINT \leq 1$ なら書き出される。)

- (7) とのプログラム内で、関数を計算した回数、いいかえれば、サブルーチンFCNGを呼び出した回数が入力で定めたNCUT を超した場合。(IER=94)
- (8) 初めて求めた σ が負になった場合。一回でも直線探索があれば、再出発。(IER = 92)
- (9) 最初のV(J, J)を求めるため2階微分係数を計算し、同じ X_J や他の X_K を変えても、5回以上もVの対角成分が零または負の値になった場合。(IER=91)

- (1) 再出発を4回やっても良い結果が得られなかった場合。(IER=93)後の(8),(9),(10)の場合には、SIMPLEX法でやり直し、その結果をDAFLEPの入力データとすればうまくゆく場合がある。
- 8. メッセージ (標準の IPRINT = 0 を主として述べる。)

関数値 F(X) の始めと終りの値は FMIN として常に書き出される。収束条件として与えた EPSF, EPSX, VTEST, NCUT 等は $IPRINT \le 0$ の場合に書き出される。 IPRINT (以下 IP と略) ≤ 0 の場合、直線探索の始まりごとに何回目の直線探索かという数、KAISU, FMIN, 対応するX(J)の値が書き出される。 $IPRINT \ge 4$ ではメッセージは一切書き出されない。その他のメッセージで自明のものは説明を省略する。以下のメッセージの終りの括弧内に記した ($IP \le NA$) の数字 NA は IP が NA 以下なら書き出されるメッセージという意味である。

- SECOND DERIVATIVE ALONG X(I=) IS NEGATIVE (IP≤1)
 I番目の変数についての2階微分が負または零で、変数空間内でF(X)が極大値か水平であることを示す。[6.0(1)参照]
- COVARIANCE MATRIX IS NOT POSITIVE DEFINITE (IP≤1)
 再出発の前にでてくれば、極小値までの推定距離のが負の場合である。再出発後に出てくれば、
 Vの対角要素を2階徴分の逆数、非対角要素を零と近似したことを示す。
- ② NS = (整数), NF = (整数), I = (expt) ($IP \le 0$) 上記 2 階微分が正でなかった場合に、I 番目のX の値に、例えばNF = 0 なら、 $0.05*3^{NS}*|X_I|$ を加えて X_I の値を大きくしたか、NF = 1 なら $-0.02*3^{NS}*|X_I|$ を加えて X_I の値を小さくしたことなどを示す。その結果のX(J)(J = 1, $\cdots N$)と対応するFMINが書き出される。($NF \ge 2$ は説明を略するが、関数の複雑な変化を意味する。)
- EXD=(数値), D(数値)D(数値)D……D(IP≤-1)
 直線探索の方向と大きさを知るには、IP≤-1とすれば、上記のJ番目のDのあとに、XD(J)の数値が入ってくる。[6.0(2)参照]
 最初に書かれたEXD=の実数は直線探索の精度を支配する量で、EXD≥1は収束しつつあることを示す。[60(3)参照]
- LIMIT ON X WAS REACHED 10 TIMES, SO RESTART (IP≤0)
 X_J が極値をとる超曲面上で、10回探索したが、収束条件を満たさないので再出発する通告。
 【6の(7)参照】 Xの上下限に片寄っている警告でもある。
- X(I=整数)=数値(LIMIT VALUE) APPEARS TWICE, SO RESTART (IP≤0)

再出発するまでの間に I 番目の X の上下限値から一度離れて、また到達したので再出発する通告。 C 6 の(7)参照。 D

© THE CONDITION ON EPSX IS NOT SATISFIED 6 TIMES, SO RESTART
(IP≤0)

関数値(の変化)についての条件 EPSF は満たしたが、Xの変化が要求(\leq EPSX)を満足せず、こういうことが5回起ったので再出発させる。(60)参照

- ▼RACE=(数値) IS. LT. VTEST=(数値) (IP≤0)
 共分散行列の対角要素の変化が小さいので計算を終了させたことを示す。・TRACEは7の(4)
 で定義された値で、VTESTは入力データである。
- © SIGMA. GT. 0.01 * EPSF, NEGATIVE GVG, OR NEGATIVE DELGAM APPEARS, SO RESTART (IP≤0)

共分散行列が正値でなくなったり、 GVG や DELGAM が負になったので再出発させる。後 二者は参考文献(4)の(4.30) に出てきたものの記号で、 GVG $=y^T \cdot H \cdot y$, DELGAM $= \Delta^T \cdot y$ を表す。

- NTRY=(整数), IER=(整数) (IP≤1)
 収束せずに不成功で終了した場合に対し、7の(7), (8), (9), (10)のそれぞれの場合に対応して、IER=94,92,91,93と書き出し、それまでの直線探索か、2階微分を求める試みかの回数がNTRY=として示される。
- ◎ 最後にV(I, J)や解の相関係数がでてくる $(IP \leq 0)$

COVARIANCE MATRIX

V(I, J)と TRACE (7の(4)) が書き出される。

 $IP \leq -2$ とすれば途中でのこれらの値も出る。

CORRELATION COEFFICIENTS:

$$\frac{V(I, J)}{\left(|V(I, I) * V(J, J)| \right)^{\frac{1}{2}}} \qquad I, J=1, \dots N$$

の値である。

GLOBAL CORRELATION COEFFICIENTS:

$$1 - \frac{1}{(V(I, I) * V^{-1}(I, I))}$$
 $I = 1, \dots N$

の値。ことで V^{-1} はVの逆行列である。

9. 使用上の注意

- (1) 元来この方法は2次関数で厳密に成立する式を基礎としているので、関数を二次関数で近似するのが不適当な領域から出発した場合、収束性が悪かったり、予想外の領域へ暴走してしまうことがある。また、変数空間内の"くぼみ"に落ちこんでしまうこともある。最終段階で、SIMPLX法の付録にあるサブルーチンXERRORを用いて、変数Xの変化を確認しておく方が安全である。
- (2) この方法は原理的には X_J の大きさによらないが、収束条件や微分係数を求めるのに、 X_J の変化率を用いているので、N個の X_J の大きさを揃えるスケーリーングを行なっておく方が このましい。
- (3) 非線型関数の最適化を取り扱う場合には、1階微分を解析的に求めないことに伴う誤差は、 関数形にもよるが、それ程深刻ではないようである。理由は、もともとこの方法は2次関数に 関して正確な方法であるのを、非線型関数に借用していることによる。4)
- (4) Xの上下限(XMAX や XMIN)附近では、その制限に引きづられて、うまく動かないようである。それ故、上下限の幅はなるべく広くとっておくのが望ましい。

IO. 備 考

文献(1)と(2)の考えに基づいたプログラムとして、東大計算機センターにHUNTER(岡井末二作成)、京大計算機センターに DAVID (星野聡作成)(後者は大阪大学に同じくDAVID として登録済)などが、登録されていて使用可能である。

II. 参考文献

- (1) W. C. Davidon, Variable Metric Method for Minimization,
 A. E. C. Research and Development Report, ANL-5990 (rev.)
 1959.
- (2) R. Fletcher and M. J. D. Powell, A Rapidly Converging

 Descent Method for Minimization, Computer J. 6 (1963), 163.
- (3) R. Fletcher, A New Approach to Variable Metric Algorithms, Computer J. 13 (1970), 317.
- (4) 小谷恒之「非線型多変数関数の極小化」 大阪大学大型計算機 センター・ニュースNa.32 (1979) pp.27.

12. 使用例

Rosenbroch の関数を例として記す。(4) (SIMPLEX法参照)。

このような簡単な場合は必要のないことだが SIMPLEX 法と組み合わせた一例を記しておく。

DAFLEP のみを用いる場合は、*印の部分を除去すればよい。

微分式を与えない(IGRAD≒1)の場合は、** 印の部分は省略してよい。 リストは次のページに写真版で記す。

作成者 : 大阪大学教養部 小谷 恒之

```
C
                                  EXAMPLE ROSENBROCK'S FUNCTION
                                  DIMENSION X(2) \times XD(2) \times XMAX(2) \times XMIN(2) \times W(30)
                                  EXTERNAL
                                                                      FCNG
                                 N = 2
                                 NW = 30
                                 READ(5,1010) (X(I), I=1,N)
                                  JOKEN=0
                                 KAISU = 0
                                 EPSF = 1.0E-7
                                 EPSX = 1.0E-4
                                 VTEST=0.01
                                 IPRINT = 0
                                 IGRAD=1
                                  EXTERNAL FCN
                                 EPSFS = 0.3
                                 EPSXS = 1.5
                                 0 = ax
                                  JPRINT = 0
                                 NMOD = 5
*
                                 NRPT = 0
                          5 CONTINUE
                                 NCUT = 50
                                 ICONT = 0
                                 CALL SIMPLX(FCN,N,X,EPSFS,EPSXS,ICONT,NCUT,JXD,XD,JOKEN,XMAX,
                                                XMIN, JPRINT, NMOD, FMIN, KAISU, WONW)
                                 DO 20 K=1.N
                      20 \times D(K) = 0.001
                                 NCUT = 400
                                 ICOVAR=0
                                 CALL DAFLEP(FCNG, N, X, EPSF, EPSX, VTEST, NCUT, XD, JOKEN, XMAX, XMIN,
                                                IPRINT, IGRAD, FMIN, KAISU, ICOVAR, WONW)
                                 IF(ICOVAR .GE. 1) GO TO 30
NRPT = NRPT + 1
                                  IF(NRPT .LE. 1) GO TO 5
 *
                       30 M = 1
                                 VARF = 0.03
                                 CALL XERROR (FCN, FMIN, N, M, X, W, NW, VARF)
                                  STOP
               1010 FORMAT( 2F10.5)
                                 END
                                  SUBROUTINE FCNG(NoxoFoGoIGRAD)
                                  DIMENSION X(N) , G(N)
                   FCA =10.0*(X(2)-X(1)*X(1))
 ** ---
                                  FCB
                                                   =1.0=X(1)
                                  F = FCA+FCA + FCB+FCB
                                  RETURN
                       10 \text{ G}(1) = -400 \text{ o} + (X(2) - X(1) + 2) + X(1) - 2 \text{ o} + (1 \text{ o} - 2) + (1 \text{ o} - 2
   **
                                  G(2) = 200.0 * (X(2) = X(1) * 2)
                                  RETURN
                                  END
                                SUBROUTINE FCN(NoxoFMIN)
                                DIMENSION X(N) oG(30)
  *
                                L = N
                                CALL FCNG(LoxoFMINoGoQ)
                                RETURN
                                END
```

分類コード	関数の極小化(POWELLの最小二乗法)		
	単精度	CALL	POWLS (SUMSQ, N, M, X, EPSF, EPSX,
C 7			NCUT, XD, JOKEN, XMAX, XMIN, IPRINT,
			FMIN, KAISU, IER, W, NW)

MINIMIZATION OF FUNCTION BY THE POWELL METHOD

I. 目 的

関数が二乗の和の形、つまり

$$F(X) = \sum_{l=1}^{M} \{ f^{(l)}(X) \}^2$$

の形をとる場合に、その性質を利用して、関数の極小値を求める。ここでXはN個の変数をベクトル形式で表した。この方法では、最初に一回だけN成分毎に、関数の偏微分係数を求めるが、形式的には偏微分係数を必要としない方法である。

関数値の変化とXの各成分についての変化率の最大値とが、それぞれ与えられた上限値より小さくなるまで、このプログラム内で自動的に反復される。

なお、XのN個の各成分の上下限をそれぞれ与えることもできるようになっている。

2. 入力データ

(2-I) SIMPLX と異なるもの。

$$M$$
………関数 $f^{(l)}$ の個数で保存される。(χ^2 -fit ではデータ数)

XD …… 1階の偏微分係数を求めるため、N個の変数 $X_J=X$ (J)(J=1,...,N)の変化率を与える。すなわち、 X_J の変化量を $\dot{\it \Delta}_J$ とし、関数 F(X)の一階偏微分係数を

$$\frac{\partial F}{\partial X_{I}} = \frac{F(X + \Delta_{I}) - F(X)}{\Delta_{I}}$$

で求める際に、差分△」を

$$\Delta_I = XD(J) * X_I$$

 $X_J=0$ の場合は、 $\triangle_J=XD$ (J) ととる。この XD は実数型 1 次元配列で、大きさは XD(K),K \ge N,保存される。

特別の事情がなければ、 $XD(J)=10^{-4}(J=1,\dots,N)$ でよい。XDが

 $0.01(10^{-5})$ より大きい (小さい)場合はプログラム内で自動的に 0.01 (10^{-5})までとされる。(6 $\mathcal{O}(5)$ の注意参照)

IER ……入力が IER = 1 の場合:

関数値が EPSF(後述) L b小さくなれば、X O値と無関係に計算は停止される。

入力が I ER ×1の場合:

SIMPLEXの場合と同じく、EPSFやEPSX等の停止の条件に従うことを指示する。(7. 停止の条件参照)

IERは保存されず、出力の内容は3. 出力データを見よ。

NW……作業領域の次元数。(4. 作業領域参照。)

計算すべき関数……

F(X)と $f_K(X)$ を定義するサブルーチンとして、使用者が作成する SUBROUTINE SUMSQ(N, M, X, F, FF, L) DIMENSION X(N), F(M, 3), FF(3)

が必要である。ことでLはプログラム内で指定されてくる数で、L=1, 2, 3。 の数値がとられる。関数との対応は

$$F(X) \rightarrow FF(L)$$

$$f_{\nu}(X) \rightarrow F(K, L)$$

である。これらのM個のF(K, L)(K=1, ..., M)とFF(L)の定義式をサブルーチンSUMSQは含んでいなくてはならない。

との POWLS を CALLするプログラムで SUMS Q に対し EXTERNAL 宣言文 が必要である。 EXTERNAL文, CALL POWLS 文, SUBROUTINE SUMS Q の 3 箇所にでてくる SUMS Q は、使用者に便利な文字で揃えてあれば よく、 SUMS Q にこだわらなくてよい。 (12. 使用例参照)

(2-2) SIMPLX と同じもの。

$$\begin{bmatrix} N \\ X \end{bmatrix}$$
 ------ SIMPLX と同じ。

EPSF } …… SIMPLX と同じ内容だが、数値は小さくしてよい。(9. 使用上の注意参照。) EPSX NCUT …… SIMPLX と同じ。実行上、(NCUT+N)回程度まで関数値を計算すること がある。

JOKEN XMAX XMIN SIMPLXと同じ。

IPRINT IPRINT = 0 が標準形で、IPRINT ≥ 4 ではなにも書き出さないなど、 SIMPLX と同じ内容だが、LMODにあたるものはない。直線探索の始まりで の関数値FMIN,対応するX, それまでのKAISUとが、IPRINT ≤ 0 なら、書き出される。

KAISU …… SIMPLX に同じ。

3. 出力データ

FMIN …… F(X)の極小値で実数型。

XFMINを与えるX(J)の値。

IER …… 条件を満す解が得られれば IER = 1, 不成功の場合には IER = -1。

4. 作業領域

W……作業用として用いられる実数型一次元配列で、大きさはW(K), $K \ge NW$ 。

NW……作業領域の大きさをきめる整数で

 $NW \ge 2 * N * (N+2) + M * (N+3)$

ととる。Mの値によるが、SIMPLX よりかなり大きくなるので注意が必要。

5. 使用サブルーチンと禁止文字群

入力データに記したように、関数 F(X)と $f_K(X)$ を定義するサブルーチン SUM SQ は使用者が作成するが、それ以外にサブルーチン KPOWと MATINVとが内蔵されているので、 これら三組の文字の組合せは使用者のサブルーチン名、共通ブロック名として使用できない。

6. 計算法の概略

この方法の基本的な考えは POWELL (1) によるものである。

本プログラムは筑波大学の小柳義夫によって1972年に作成されたものを、小谷が修正したものである。(以下*印を付した項目が小谷の修正である。)本プログラムでなされた、文献(1)にない特別な工夫の要点は下記のような諸点である。(2)

(1) 最初の直線探索の方向を決めるのに最大傾斜の方向に重みをつけて、共分散行列 (H)の対角要素を1.005倍してある。

この際、 $f_{\kappa}(\mathbf{X})$ の 1階微分係数を求める必要があるが、規格化をして、パラメーターのス

ケールの選び方による不定性をなくしてある。

- (2) * 関数 F(X) の X_J に関する微分係数が零になるような高原型の変化の場合、 X_J の値を-0.18 倍から 1.7 倍の間で微分係数が零にならない X_J の値を探して、それを出発の X_J とする。この範囲に X_J が見出だせない場合、J番目のXの依存性なしとしてプログラムは停止する。
- (3)* 関数値を極小にする直線探索の精度は、 X_J の変化率が1 %位から初め、極値に近づくと、精度があがるようにした。
- (4)* X_J を変化させて、その値が指定された上下限($XMAX_J$ や $XMIN_J$)の外へ出た場合には、 X_J =一定の超曲面上で、 X_J 以外の変数への依存性が調べられる。(この状態で10 回以上直線探索がくり返されると再出発させる。) X_J の値が上下限間にもどれば、 X_J も変化させるのはもちろんである。しかし、再び上下限外へ出ると直ちに再出発させる。
- (5)* 直線探索の方向と大きさを与えるベクトルの成分が異常に大きくなることがある。例えば、 N個の共役方向が同一方向に近い場合、関数値 Fが急激に変化する絶壁の近傍の場合、また、 Fの X 依存性が弱い 高 原 型の変 化のような場合が考えられる。方向だけ尊重して 5 回直線探索が試みられるが、この状態がさらに続くようだと再出発させる。この再出発では、微係数を 求める X Dは 2 倍にとられ、微係数 も $\left\{F\left(X+\Delta_{I}\right)-F\left(X-\Delta_{I}\right)\right\}/2$ Δ_{I} と 特度をあげて計算される。
- (6)*関数 Fの変化が EPSF より小さくなると、Xの変化率の最大値が EPSX より小さいかど うかが検査される。この関門を5回までで合格しないと、始めから再出発させる。
- (7) POWELLの論文の(4) ~(15)式 による近似の改良はかえって、上述(5)の異常発生の誘因になるようなので採用しないことにした。

とのため、共役方向の修正が同一方向について、続けて行なわれる確率は高くなっている欠 5はある。

(8)*直線探索は少なくともN回は原則として実行される。このNは変数の数である。

7. 停止の条件

(7-I) 収束して終了(成功, IER=1)

- (1) 関数値が EPSF 以下になれば、入力で IER=1 の場合は停止し、 $IER \succeq 1$ の場合、 0.01*EPSF 以下なら停止、そうでなければ関門(3)へ移る。
- (2) 直線探索を始める前の関数値と変数の値をそれぞれFMIN, X(J) とし、探索後の対応する値をFB, Y(J) とすると、

FMIN - FB < EPSF

ならば関門(3)へ移る。

(3) (1)か(2)を通過後、次の条件が満された場合に停止する。

$$\max_{1 \le J \le N} |1 - \frac{Y(J)}{X(J)}| < \text{EPSX}$$

(4) 3 回再出発したが、直線探索で X_J が変化しても、関数値が不変である場合。(途中の極小値で停滞している可能性があるので注意を要するが、極小値に近づくと起こり易いので成功としてある。)

(7-2) 収束せずに終了(不成功,IER=-1)

- (5) とのプログラム内で関数を計算した回数、いいかえれば、サブルーチンSUMSQを呼び出した回数が入力で定めたNCUTを超した場合。
- (6) 出発点で、あるX, について1階微分係数が零であった場合。(6の(2)参照)
- (7) 再出発を3回やっても良い結果が得られなかった場合。 後の二者の場合、SIMPLEX法でやり直し、その結果をPOWLSの入力データとすれば、うまく作動する場合が多い。
- 8. メッセージ (標準のIPRINT=0を主として述べる。)

関数値 F (X) の始めと終りの値は FMIN として常に書き出される。収束条件として与えた EPSF, EPSX, NCUTは $IPRINT \leq 0$ の場合に書き出される。

IPRINT (以下 IP と略) \leq 0 の場合、直線探索の始まり毎に、何回目の直線探索かという数字、KAI SU,FMIN,対応するX(J)の値が書き出される。その他のメッセージ中自明のものの説明は省略する。以下のメッセージの終りの括弧内に記した($IP \leq NA$)の数字NAはIPがNA以下なら書き出されるメッセージという意味で、標準のIP = 0以上についてのみ記す。

I P≥4ではメッセージは一切でてこない。

◎ F(X) IS FLAT ALONG X(整数)

VALUE OF X(1=整数)WILL BE CHANGED (IP≤1)

I番目のXについて1階微分が0だったので、X(I)の値を変化させて試みた。(6 O(2)参照)

- F(X) DOES NOT DEPEND ON X(I=整数)(IP≤2)
 I番目のX(I)を前記のように変化させてみたが、有限の1階微分が求まらなかった。
- HESSIAN MATRIX IS NOT TOSITIVE—DEFINITE (IP≦1)ヘシアン行列が正値でなかったので、次のメッセージも記して再出発。
- © THE NORMAL EQ IS SINGULAR, SO RESTART (IP≤1)

上述の場合、および修正した共分散行列の補正項が異常に大きい場合で、再出発。

- LIMIT ON X WAS REACHED 10 TIMES, SO RESTART (IP≤0)
 Xの上下限に達し、その値を固定して10回探索したので、再出発する(6の(4)参照)。Xの上下限に片寄っている警告。
- X(J)=数値(LIMIT VALUE) APPEARS TWICE, SO RESTART (IP≤1)
 Xの上下限に一度到達し、その後との極値から離れたが、またとの極値へ達したので再出発。
- F-VALUE IS TOO LARGE, SO RESTART (IP≤0)
 直線探索の方向か大きさが異常で、関数の値が前回より30倍以上になることが5回続いたので再出発させる。再出発後は微分係数の求め方が変えられる。(6の(5)参照)
- © CONDITION ON X WAS NOT SATISFIED 4 TIMES, SO RESTART
 (IP≤0)

関数値(の変化)の条件 EPSF は満たしたが、Xの変化が要求 (\leq EPSX) を満足せず、こういうことが 5回起ったので再出発させる。(6の(6)参照)

- FAILS TO FIND THE SMALL F, LT. FMIN, SO RESTART (IP≤0)
 直線探索を少くともN回試みたが、出発値のFMINより小さい関数値を求められなかったので、
 再出発。
- MINIMUM POSITION R(L)=数値 IS EQUAL TO THE 2 ND MINIMUM R(LMIN)=数値, SO RESTART (IP≤0)
 直線探索で関数値の極小値と次の小さい値とが計算機の誤差のため偶然一致したことを示し、再出発させる。R(L)とR(LMIN) は直線探索を行なうベクトルの方向で二つの関数値の位置を示す係数である。

9. 使用上の注意

(1) 元来この方法は2次関数で正しく成立する式を基礎としているので、関数を2次関数で近似するのが不適当な領域から出発した場合、収束が悪かったり、予想外の領域へ暴走してしまうととがある。また、変数空間の"くぼみ"に落ち込んでしまうと、脱出しにくいようである。それ故SIMPLX 法と併用し、特にその付録にあるサブルーチンXERROR を用いて、最終の

変数Xの変化を確認しておく方が安全である。

- (2) この方法は原理的には X_J の大きさによらないが、収束条件や微分係数を求めるのに、 X_J の変化率を用いているので、N 個の X_J の大きさを揃えるスケーリングを行なっておく方がとのましい。
- (3) 微分係数を求めないでというのが、原著者⁽¹⁾がこの方法の特色としている点であるが、実質的には直線探索の極小値附近での差分を用いて、新しい共役方向を選んでいる。⁽²⁾2次関数に適した方法を借用しているという意味で、解析的に微分係数を求めないことによる欠点は打ち消されている。
- (4) Xの上下限附近ではその制限に引きづられて、 うまく動作しないことがある。それ故、上下 限の間隔はなるべく大きくとっておくのが望ましい。

10. 備 考

文献(1)の考えに基づいたプログラムとして東大計算機センターにPOW1(小柳義夫作成), 日立のサービス・ライブラリーにPLSMが登録されていて、使用可能である。蛇足ながら、大阪大学のライブラリーに入っているPOW(星野聡作成)は同じPOWELLの提案だが別の考えに基づくものである。(2)

II. 参 考 文 献

- (1) M.J.D. Powell, A Method for Minimizing a Sum of Squares of Nonlinear Functions without Calculating Derivatives, Computer Journal 7.(1965), 303.
- (2) 小谷恒之「非線型多変数関数の極小化」 大阪大学大型計算機センター・ニュース No. 3 2, (1979), pp. 27.

I 2. 使 用 例

Rosenbrock の関数(SIMPLEX法参照) を例として記しておく。⁽²⁾ このような簡単な場合には必要のないことだが、SIMPLEX法とXERRORとを組み合せた例 を記す。POWLS のみを用いる場合は、*印の部分を除去すればよい。

リストは次のページに写真版で記す。

作成者 : 大阪大学教養部 小谷 恒之

```
EXAMPLE ROSENBROCK'S FUNCTION
C
      DIMENSION X(2) \neq XD(2) \neq XMAX(2) \neq XMIN(2) \neq W(30)
      EXTERNAL SUMSQ
      COMMON/DATA/ MD
      N = 2
      M = 2
      NW = 30
      READ(5,1010) (X(I),I=1,N)
      EPSF = 1.0E ≈7
      EPSX = 1,0E-4
      JOKEN = 0
      KAISU = 0
      IPRINT = 0
      EXTERNAL FON
      MD= M
      EPSFS = 0.3
      EPSXS ≈ 1.5
      JXD = 0
      JPRINT = 0
*
      NMOD = 5
      NRPT = 0
    5 CONTINUE
      NCUT = 50
      ICONT = 0
      CALL SIMPLX(FCN,N,X,EPSFS,EPSXS,ICONT,NCUT,JXD,XD,JOKEN,XMAX,
           XMIN, JPRINT, NMOD, FMIN, KAISU, W, NW)
      DO 20 K=1.N
   20 \text{ XD(K)} = 0.0001
      NCUT=400
      IER = 0
      CALL POWLS(SUMSQ,N,M,X,EPSF,EPSX,NCUT,XD,JOKEN,XMAX,XMIN,
           IPRINT, FMIN, KAISU, IER, W, NW)
      "If(IER "GE" 1) GO TO 30
      NRPT = NRPT + 1
      IF(NRPT ,LE. 1) GO TO 5
*{30 M = 1
      VARF = 0.03
      CALL XERROR (FCN, FMIN, N, M, X, W, NW, VARF)
      STOP
 1010 FORMAT( 2F10.5)
      END
      SUBROUTINE SUMSQ(N,M,X,F,FFF,L)
      DIMENSION X(N) \rho F(M \rho 3) \rho F F(3)
      F(1/L) = 10.0 * (X(2) + X(1) * X(1))
      F(2_{\ell}L) = 1.0 - X(1)
      FF(L) = F(1_0L) *F(1_0L) + F(2_0L) *F(2_0L)
      RETURN
      END
      SUBROUTINE FCN(N,X,FMIN)
      DIMENSION X(N) \nearrow F(1000,3) \nearrow FF(3)
      COMMON/DATA/ MD
 X
      L = N
      CALL SUMSQ(L,MD,X,F,FF,1)
      FMIN = FF(1)
      RETURN
      END
```