論　文

微分式を用いない最小2乗法について*

吉　田　二　朗**

Abstract

In this paper, the method of calculating the minimizing direction due to DFP(Davidon-Fletcher-Powell) is described. This is used in the least squares method for an over-determined system. I proved that if the gradient was calculated by Powell's method (1964) in the space formed by the minimizing directions (shorten as the method in the direction's space), it was possible to do also in the direction's space on the decision of the directions.

I calculated numerically some test functions and compared them with some descent methods. As a result, I think that this method may be useful in reducing the computing time for the minimization function by which the jacobian matrix can be calculated with reasonable accuracy by the method in the direction's space.

1. まえがき

本論文は，優決定系に対する最小2乗法に傾斜法DFP[1]法を改良した方法を提案している．このDFP法の背景について述べると次のようである．制約条件のない最小化問題の解き方は，主に傾斜法と直接探索法にわけられる．ある点から方向を決定するとき，前者はその点，あるいはその前の点の勾配を用いて決定するのに対し，後者は勾配を用いない方法である．傾斜法には方向の計算法によって，2次導関数の逆行列による方法（GN法と略す），最大傾斜法，共役傾斜法，等がある．共役傾斜法は，2次導関数の逆行列の展開形のような方法であるが，以下に述べる方法によるので，非線形問題に対して有用である．共役傾斜法は，関数を2次形式で近似し，その2次導関数行列に関して，互いに共役な方向ベクトルの組をつくりながら，各組の方向の直線上の極小点をつぎつぎに求め，最後に最小点に到達せしめる方法である．さて，共役なベクトルをつくる方法は，色々提案されているが，DFPの方法を，本論文ではとりあげる．勾配計算法として，Powell[2]の提案した方向勾配法を用いているが，本論文では，方向の計算についても方向空間内で求めるという，新しい方法を提案する．この方法は，従来のDFP法とくらべて明らかに計算量が減少すると考える．1方向の直線上の最小点を求める方法は，計算式によってではなく，DFP法に述べてあるように，直線探索法によっている．

2. 最小化関数と空間

2.1 最小化関数

最小化関数 F は，

$$F=f^t\cdot f \qquad (1)$$

である．ここで，$f^t=(f_1,\cdots,f_m)$，$x^t=(x_1,\cdots,x_n)$ 優決定系だから，$m\geq n$.

2.2 元空間と方向空間

（1） 元空間（×-space).

アルゴリズムステップの逐行に不変な直交座標系を，元空間または，×-spaceとよぶことにする．記号の右下に × と書くとき，×-space 内で表わした量とする．

（2） 方向空間（T-space).

n 個の独立な方向ベクトル $t_x{}^{(1)}\cdots t_x{}^{(n)}$ を軸とする座標系で，方向空間または，T-space とよぶことにする．記号の右下に T と書くとき，T-space 内で表わ

* The Least Squares Method Without Calculating Derivatives by Ziro YOSIDA (Computer Room, Faculty of Engineering, Shizuoka Universiy).

** 静岡大学工学部電子計算機室

した量とする．また，T は，方向行列の意味ももち，

$$T=\frac{\partial x}{\partial t}=(t_x^{(1)}, \cdots, t_x^{(m)})$$

で表わすことができる．T は，前ステップの空間 T_0 の適当な列 im (3.2（2）参照) を方向ベクトル t_x でおきかえた行列で，次の式で表わせる．

$$T\leftarrow T_0+(t_x-t_x^0)\cdot e_{im}{}^t \qquad (2)$$
$$t_x^0\leftarrow\kappa_{im}(T_0)$$

3. 勾配計算法

2.1 の最小化関数 F の勾配 g は，f のヤコビ行列 A を用いて，次の式で表わせる．

$$g=2\cdot A^t\cdot f \qquad (3)$$

ここで，$A=\partial f/\partial x$．従って，A の求め方によって，以下の2つの計算法がある．

3.1 元空間法（×-space 法）

座標軸の各軸方向の摂動によって求める方法で，次の式で表わせる．

$$A_{xij}\leftarrow(f_i(x+\lambda e_j)-f_i(x))/\lambda,$$
$$i=1, m,\quad j=1, n. \qquad (4)$$

ここで，λ は適当に小さいスカラー．

$$g_x\leftarrow 2\cdot A_x{}^t\cdot f \qquad (5)$$

3.2 方向空間法（T-space 法）

（1） 方向勾配法

アルゴリズム上のあるステップで，方向ベクトル t_x の直線上の f の勾配が g_{t_x} であり，T_0-space のヤコビ行列が A_{T_0} であるとき，新しい空間 T の A_T は，A_{T_0} の im 列を g_{t_x} におきかえたものと近似的に等しい．このことを式で表わすと，

$$A_T\leftarrow A_{T_0}+(g_{t_x}-g_{t_x}{}^0)\cdot e_{im}{}^t \qquad (6)$$
$$g_{t_x}{}^0\leftarrow\kappa_{im}(A_{T_0})$$

だから，$g_T\leftarrow 2\cdot A_T{}^t\cdot f$

（2） 交換軸の選択法

Powell[1]の方法によって，交換軸 im を求める．T-space の F の勾配と，方向の各成分の積の絶対値が最大となる番目を求め，これに対応する軸を方向 t_x とおきかえ新しい軸とする．（式（2）と同じ意味）

$$im\leftarrow\max_i(|g_{T_i}\cdot t_{T_i}|) \qquad (7)$$
$$\kappa_{im}{}^0(T_0)\leftarrow\kappa_{im}{}^0(T)$$
$$\kappa_{im}(T)\leftarrow t_x$$

（3） g_{t_x} の計算

Powell の方法によって，つぎのように求める．方向 t_x の直線上の f の勾配（方向勾配と略）$df/d\lambda$ は，1階の差分商を求めて，直線探索の結果が $dF/d\lambda=0$ になることを用いて，補正して求める．式で表わすと，

$$g_{t_x}{}'\leftarrow(f(x+\lambda\cdot t_x)-f(x))/\lambda$$
$$\mu\leftarrow f(x+\lambda t_x)^t\cdot g_{t_x}{}'/\|f(x+\lambda t_x)\|^2$$
$$g_{t_x}\leftarrow g_{t_x}{}'-\mu\cdot f(x+\lambda t_x)$$

（4） T-space の軸勾配法

T-space の各軸の摂動によって求める方法で，つぎの式で表わせる．

$$A_{T_{ij}}\leftarrow(f_i(x+\lambda\cdot\kappa_j(T))-f_i(x))/\lambda,$$
$$i=1, m,\ j=1, n$$

だから，$g_T=2\cdot A_T{}^t\cdot f$

4. 方向決定計算法

4.1 近似2次偏導関数行列 M の逆行列による方法

$$M_x=\frac{\partial^2 F}{\partial x^2},\quad M_T=\frac{\partial^2 F}{\partial t^2}$$

（1） ×-space 法 (Gauss-Newton 法，略して GN 法)

$$M_x\leftarrow 2\cdot A_x{}^t\cdot A_x$$
$$dx\leftarrow -M_x{}^{-1}\cdot g_x$$
$$t_x\leftarrow dx/\|dx\|$$

（2） T-space 法 (Powell の最小二乗法略して PWL 法)

$$M_T\leftarrow 2\cdot A_T{}^t\cdot A_T$$
$$dt\leftarrow -M_T{}^{-1}\cdot g_T$$
$$t_T\leftarrow dt/\|dt\|$$

4.2 DFP 法

（1） DFP の ×-space[2]法

$t_x\leftarrow -H_x\cdot g_x$，ここで H_x は，つぎのようである．

$$A_x\leftarrow\frac{dx\cdot dx^t}{dx^t\cdot dg_x},\quad B_x=\frac{H_x{}^0\cdot dg_x\cdot dg_x{}^t\cdot H_x{}^0}{dg_x{}^t\cdot H_x{}^0\cdot dg_x} \qquad (8)$$
$$H_x\leftarrow H_x{}^0+A_x-B_x$$

（2） DFP の T-space 法

ここが本論文で提案するところである．

$$t_T\leftarrow -H_T\cdot g_T$$

ここで，H_T は，つぎのように求める．

$$\left.\begin{aligned} &g_T{}^0\leftarrow(T_0{}^{-1}\cdot T)^t\cdot g_{T_0}\\ &dg_T\leftarrow g_T-g_T{}^0\\ &H_T{}^0\leftarrow(T^{-1}\cdot T_0)\cdot H_{T_0}\cdot(T^{-1}T_0)^t\\ &A_T\leftarrow\frac{dt\cdot dt^t}{dg_T{}^t\cdot dt},\quad B_T\leftarrow\frac{H_T{}^0\cdot dg_T\cdot dg_T{}^t\cdot H_T{}^0}{dg_T{}^t\cdot H_T{}^0\cdot dg_T}\\ &H_T\leftarrow H_T{}^0+A_T-B_T \end{aligned}\right\} \qquad (9)$$

証明

$$dt=T^{-1}\cdot dx$$

$g_T=T^t\cdot g_x$, $g_{T_0}=T_0{}^t\cdot g_x{}^0$, $dg_T=T^t dg_x$,

$H_x=T\cdot H_T\cdot T^t$, $H_x{}^0=T_0\cdot H_{T_0}\cdot T_0{}^t$ より, A_T, B_T, $H_T{}^0$, H_T を式 (9) へ代入することにより式 (8) の $\times$-space の式が導かれる. 逆に, $\times$-space の式から, 上式の関係を用いて, T-space の式 (9) が導かれる.

5. DFP の T-space 法の簡単化

$$v=T_0{}^{-1}\cdot(t_x-t_x{}^0),$$

$$\mu=-\frac{1}{1+v_{im}},$$

$$\nu=v^t\cdot g_{T_0}$$

とおく.

5.1 変換行列に関する計算

(1) T^{-1} の逆行列を用いない計算

$$T^{-1}\leftarrow(I+\mu\cdot v\cdot e_{im}{}^t)\cdot T_0{}^{-1}$$

(2) $T^{-1}\cdot T_0$, $T_0{}^{-1}\cdot T$ の計算

$$T^{-1}\cdot T_0=I+\mu\cdot v\cdot e_{im}{}^t$$

$$T_0{}^{-1}\cdot T=I+v\cdot e_{im}{}^t$$

5.2 dg_T, $H_T{}^0\Leftarrow H_{T_0}$ の計算

$$dg_T\leftarrow g_T-g_{T_0}-\nu\cdot e_{im}$$

$$H_T{}^0\leftarrow H_{T_0}+\mu\cdot\kappa_{im}(H_{T_0})\cdot v^t+\mu\cdot v\cdot\rho_{im}(H_{T_0})+\mu^2\cdot H_{T_0,im,im}\cdot v\cdot v^t$$

$H_T{}^0\Leftarrow H_{T_0}$ の計算は掛算回数をつぎのようにして減ずることができる.

$$\left.\begin{aligned}\omega^t&\leftarrow\rho_i(H_{T_0})+\mu\cdot v_i\cdot\rho_{im}(H_{T_0})\\ \rho_i(H_T{}^0)&\leftarrow w^t+\mu\cdot w_{im}\cdot v^t\end{aligned}\right\}i=1,n$$

5.3 A_T の計算

方向 t_T は, T-space の軸の 1 つである. だから dt は, im 成分以外は 0 のベクトルである.

$dt=\Delta t\cdot e_{im}$ とかける. Δt は, スカラー.

$$A_T=\frac{dt\cdot dt^t}{dg_T{}^t\cdot dt}=\frac{\Delta t}{dg_{T,im}}e_{im}\cdot e_{im}{}^t$$

6. T-space の DFP 法 (FLTSQ) のフローチャート

6.1 メインルーチン

フローチャートは, **Fig. 1** である. フローチャートで使われている記号の説明は, つぎのとおり.

ε: 絶対精度

ΔF: 前ステップと今ステップの関数値の差

$\varepsilon\varepsilon$: ΔF の精度

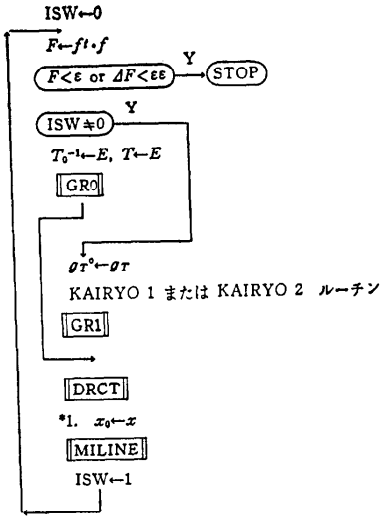
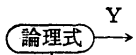


Fig. 1 Main routine of FLTSQ.

exp 1 or exp 2 : 論理式 exp 1 と exp 2 の論理和.



; 論理式の値が真のとき Y の矢印のある方向へ制御が渡る.

A←B : 付録 1 参照.

ENTRY エントリ名 : 他のルーチンからこのエントリへ制御を渡す.

＊1 : 他のルーチンの RETURN 1 のとき, ここへ制御が渡る.

E : 単位行列

GR 0 : $\times$-space の勾配計算ルーチン

GR 1 : 方向勾配計算ルーチン

DRCT : T-space の方向計算ルーチン

MILINE : 直線探索ルーチン名

流れ線と矢印がないとき上から順に実行される.

R : R は, ルーチン

6.2 GR 0 ルーチン

Fig. 2, **3.1** 参照

6.3 GR 1 ルーチン

Fig. 3 (次頁参照), **3.2** 参照.

6.4 DRCT ルーチン

Fig. 4 (次頁参照), **4.2** の (2) 参照.

$\kappa_j(A_T)\leftarrow(f(x+\lambda e_j)-f(x))/\lambda$, $j=1, n$

$g_T\leftarrow 2\cdot A_T{}^t\cdot f$

RETURN

Fig. 2 Routine of GR 0.

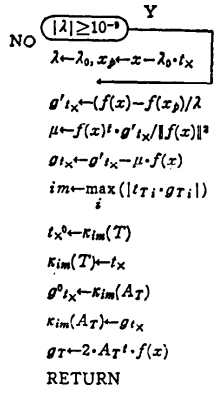


Fig. 3 Routine of GR 1.

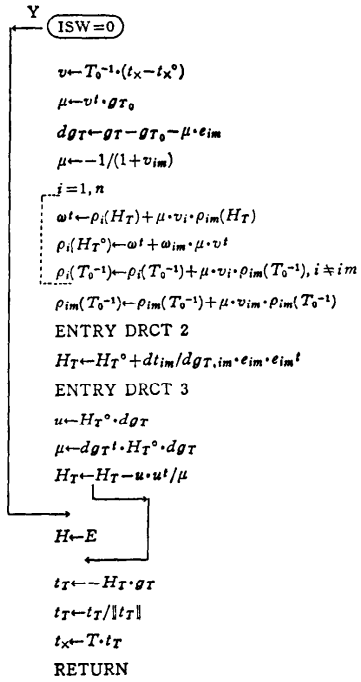


Fig. 4 Routine of DRCT.

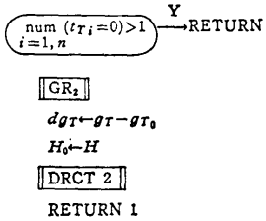


Fig. 5 Routine of KAIRYO 1.

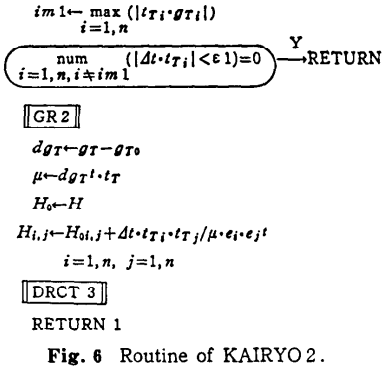


Fig. 6 Routine of KAIRYO 2.

6.5 MILINE ルーチン[3)]

直線探索ルーチンであり，3点2次近似式を用いた．フローチャートは省略する．

7. 比較したアルゴリズム

Table 1 のとおりである．

ここで提案しているアルゴリズムは FLTSQ である．なお，( )内は説明してある章節．

⑤ の NTNGS は古典的な Gauss Newton 法に直線探索が追加されている．

⑥ は微分式を含む従来の DFP 法であるが，これと ③ とどの程度の差があるか分数関数について調べた．このことについて，**9.3** に記してある．

8. テスト関数

8.1 乱数行列の2次形式

行列 A は，0から1までの乱数で，非対角要素だけを0.5倍した要素をもつ正方行列とする．

$$f=A\cdot x$$

$$F=f^t\cdot f$$

$$x_0=(10,\cdots,10),\quad \hat{x}=(0,\cdots,0),\quad \hat{F}=0$$

x_0: 初期点，F_0: 初期値

$\hat{x}$: 最小点，$\hat{F}$: 最小値

初期値は，下記のようである．

$n=10$ のとき，$x_0=(\overbrace{10,\cdots,10}^{10\text{ 個}})$，$F_0=0.88771\times 10^4$

Table 1 Compared algorithms.

番号	アルゴリズム名	勾配計算法	方向決定計算法
①	FLTSQ	T-Space 方向勾配 (3·2)	T-Space DFP (4·2(2))
②	FLTSQ 1	〃 〃	×-Space DFP (4·2(1))
③	FLTSQ 2	×-Space 軸勾配 (3·1)	〃 〃
④	PWLSQ	T-Space 方向勾配 (3·2)	PWL 法 (4·1(2))
⑤	NTNGS	×-Space 軸勾配 (3·1)	GN 法 (4·1(1))
⑥	FLTPL	微分式による	×-Space DFP (4·2(1))

$n=20$ のとき, $x_0=(\overbrace{10, \cdots, 10}^{20\text{個}})$, $F_0=0.60856\times10^5$

$n=30$ のとき, $x_0=(\overbrace{10, \cdots, 10}^{30\text{個}})$, $F_0=0.18790\times10^6$

8.2 標準テスト関数[4]

(1) Rosenbrock の関数

$x_0=(-1.2, 1)$, $F_0=24.2$

$\hat{x}=(1, 1)$, $\hat{F}=0$

(2) 3次の谷の関数

$x_0=(-1.2, 1)$, $F_0=749.04$

$\hat{x}=(1, 1)$, $\hat{F}=0$

(3) Beale の関数

$x_0=(0.1, 0.1)$, $F_0=6.6149$

$\hat{x}=(3., 0.5)$, $\hat{F}=0$

(4) BOX の関数

$x_0=(0, 10, 20)$, $F_0=1031.2$

$\hat{x}=(1, 10, 1)$, $\hat{F}=0$

$\hat{x}=(3.1664, 3.1664, 0.51917\times10^{-6})$, $\hat{F}=0$

この結果は，文献 4) にない.

(5) 変形 BOX の関数

$f_i=(e^{-x_1\cdot y_i}-e^{-x_2\cdot y_i})-x_3\cdot(e^{-y_i}-e^{-10\cdot y_i})$

$y_i=1, 2, \cdots, 10$ 標準関数は $y_i=0.1,\ 0.2,\ \cdots\ 1.0$

$F=f^t\cdot f$

$x_0=(0, 10, 20)$, $F_0=49.318$

$\hat{x}=(1, 10, 1)$, $\hat{F}=0$

(6) Enzyme の関数

$x_0=(0., 0., 0., 0.)$, $F_0=0.14841$

$\hat{x}=(0.1928, 0.1916, 0.1234, 0.1362)$

$\hat{F}=3.075\times10^{-4}$

(7) (6)の x_0 の値を変えたとき.

$x_0=(0.25, 0.39, 0.415, 0.39)$, $F_0=0.53188\times10^{-3}$

$\hat{x}=(0.1928, 0.1916, 0.1234, 0.1362)$

$\hat{F}=3.07\times10^{-4}$

8.3 分 数 関 数[5]

角周波数 ω_i において，変数 x の関数を y_i，仕様を y_{si} とすると，

$$y_i(x)=20\cdot\log|T_i(x)|,\quad i=1, m$$

$$f_i=y_i(x)-y_{si},\quad i=1, m$$

$$F=f^t\cdot f$$

F を最小にする x を求める.

ここで,

$$T_i(x)=\frac{\sum_{k=1}^{n_n} a_k\cdot s_i^{k-1}}{\sum_{k=1}^{d_n} b_k\cdot s_i^{k-1}}$$

x は係数 $\{a_k\}$, $\{b_k\}$ から選ばれた変数, $s_i=j\cdot\omega_i$, $j\sqrt{-1}$.

数値計算例

$x=\{a_1, a_2, a_3, a_4, a_5, a_6\}$

$b_1=1$, $b_2=0$, $b_3=0.5$, $b_4=0$, $b_5=0.0625$

仕様 f_{si} は (ω_i, f_{si}) **Table 2** のように与える.

$x_0=(1, 1, 1, 1, 1, 1)$, $F_0=3,354$

$\hat{x}=(1.719, 8.0745, 1.9031, 19.573, 1.2726, 8.9202)$

$\hat{F}=105.62$

Table 2 Specification of a fractional rational function.

(0, 6), (0.2, 6), (0.6, 6), (0.6, 6), (0.8, 6), (1, 9), (1.1, 14), (1.2, 18), (1.4, 27), (1.6, 40), (1.95, 95.5), (2.05, 97.4), (2.2, 78), (2.6, 65), (2.8, 63), (3., 62), (3.2, 61), (3.4, 61), (3.8, 60), (4., 60)

Table 3 Quadratic form with a random matrix.

アルゴリズム名	元 数	方向決定回数	関数呼出し回数	精 度	計算時間(秒)
FLTSQ	10元	10	215	0.41505×10^{-14}	5.9
	20元	20	218	0.81346×10^{-12}	26.1
	30元	30	231	0.34466×10^{-9}	74.9
FLTSQ1	10元	10	215	0.40684×10^{-14}	6.2
	20元	20	218	0.79812×10^{-12}	28.0
	30元	30	231	0.31960×10^{-9}	81.9
FLTSQ2	10元	10	215	0.95403×10^{-15}	6.4
	20元	20	218	0.6887×10^{-13}	38.5
	30元	30	231	0.20037×10^{-10}	144.4
PWLSQ	10元	1	102	0.30115×10^{-21}	2.2
	20元	1	102	0.13952×10^{-20}	8.3
	30元	1	103	0.17719×10^{-20}	20.4
NTNGS	10元	1	102	0.30115×10^{-21}	2.3
	20元	1	102	0.13952×10^{-20}	8.3
	30元	1	102	0.17719×10^{-20}	20.7

9. テスト関数の結果と検討

9.1 FLTSQ による乱数行列の2次形式の結果

Table 3 のように予想どおり，従来の FLTSQ2 よりも，元数が多くなるに従って，処理速度が早くなり，30 元では2倍に達する.

9.2 標準テスト関数，分数関数の結果

Table 4, **Table 5** (次頁参照) のとおり. 後に **11.** で述べるように，DFP 法がすぐれており，また，**10.** で述べるわずかな改良によって FLTSQ が有効なアルゴリズムになった.

計算値のリストは，関数値が 10^{-6} 以下になったとき，または，今ステップと前ステップの関数値の差が 10^{-5} 以下になったときについて記してある.

9.3 分数関数の各アルゴリズムの計算時間の差について

分数関数について処理時間を調べてみると， Table

Table 4 Standard test function 1.

① FLTSQ, ② FLTSQ 1, ③ FLTSQ 2, ④ PWLSQ, ⑤ NTNGS, ⑥ FLTPL

関数名 x_0, F_0, λ　アルゴリズム番号	方向決定回数	関数呼び出し回数	精　度 F	最　終　値　x
(1) ROSEN BROCK ①, ②	28	157	0.31365×10^{-7}	(0.99984, 0.99967)
(−1, 2, 1) ③	19	115	0.97111×10^{-8}	(0.99956, 0.99903)
24.2 ④	20	120	0.60564×10^{-7}	(0.99975, 0.99951)
$\lambda=0.1$ ⑤	15	94	0.10602×10^{-6}	(1.0003, 1.0006)
(2) 3次の谷 ①, ②	15	95	0.18723×10^{-8}	(1.0000, 1.0001)
(−1.2, 1) ③	11	72	0.69184×10^{-6}	(0.99926, 0.99782)
749.04 ④	20	151	0.48713×10^{-6}	(0.99937, 0.99815)
$\lambda=0.01$ ⑤	11	81	0.27728×10^{-6}	(0.99997, 0.99990)
(3) BEALE ①, ②	11	68	0.20912×10^{-7}	(3.0000, 0.49997)
(0.1, 0.1) ③	8	59	0.55172×10^{-6}	(2.9964, 0.49915)
6.6149 ④	9	66	0.3954×10^{-6}	(3.0002, 0.50005)
$\lambda=0.01$ ⑤	7	87	0.46174×10^{-7}	(2.9998, 0.50001)
(4) BOX ①, ②	6	269	0.42531×10^{-8}	(−2.1653, 29.281, 16.678)
(0, 10, 20) ③	10	304	0.16785×10^{-6}	(3.1694, 3.1833, 0.22159×10^{-2})
1031.2 ④	23	299	0.15747×10^{-1}	(0.64409, 14.025, 1.2407)
$\lambda=0.01$ ⑤	3	215	0.57329×10^{-7}	(1.99935, 10.003, 1.0003)
(5) 変形 BOX ①, ②	9	234	0.14639×10^{-7}	(0.99980, 10.003, 0.99994)
(0, 10, 20) ③	7	220	0.24845×10^{-6}	(0.99916, 10.003, 0.99976)
49.318 ④	最初の直線探査で，タイム（アルゴリズムを比較するための十分な時間）オーバー			
$\lambda=0.01$ ⑤	〃			〃
(6) ENZYME 1 ①, ②	3	8	0.51743×10^{-1}	フローティングオーバーフロー (0.93745×10^{-1}, 0, 0, 0)
(0, 0, 0, 0) ③	4	7	0.6791×10^{-8}	(0.19245, 0.21765, 0.15879, 0.136)
0.14841 ④	最初の方向計算で M_T の逆行列がない.			
$\lambda=0.1$ ⑤	〃		M_X の逆行列がない.	
(7) ENZYME 2 ①, ②	25	90	0.31054×10^{-8}	(0.19272, 0.19665, 0.1249, 0.13876)
(0.25, 0.39, 0.415, 0.39) ③	25	91	0.31042×10^{-8}	(0.19276, 0.1940, 0.12459, 0.13704)
0.53188×10^{-3} ④	84	202	0.14072×10^{-3}	(0.23961, 0.61589×10^{-1}, 0.40398, 0.49420×10^{-1})
$\lambda=0.1$ ④	6	33	0.31322×10^{-8}	(0.19475, 0.15815, 0.12002, 0.12065)

λ は初期の λ.

Table 5 Standard test function 2.

関数	アルゴリズム番号	方向決定回数	関数呼出し回数	精　度	時間(秒)
分数	①	80	304	314.12	
	②	97	437	324.4	
	③	16	299 (395)	105.62	97.4
	④	80	508	426.15	
	⑤	17	648	105.62	181.4
	⑥	17	303	105.62	88.3

5 より，FLTSQ 2 ③ と FLTPL ⑥ とはわずかな差であり，NTNGS ⑤ は，2倍かかる．これらの処理時間の差の原因は次のようになる．FLTSQ 2 と FLTPL は，方向決定回数は同じであり，関数呼びだしは，それぞれ 345 回，303 回である．この差と微分式計算時間とが，97.4 秒，88.3 秒となって現われる．FLTSQ ① は不成功であったが，**10.** で述べる改良によって成功した．

10. FLTSQ の改良について

10.1 改良の目的

Table 4, Table 5 のように FLTSQ は，Enzyme 関数，BOX 関数，分数関数に関して解くことができなかったが，FLTSQ 2 は成功した．この理由は，ヤコビ行列の方向勾配法に問題があるからである．Enzyme 関数は，1つの方向に捕縛され，ヤコビ行列が更新されないためであり，BOX，分数関数は，直線探索の結果の揺動が大きすぎてヤコビ行列の誤差が大きくなるためである．

10.2 方向勾配法の問題点と対策

ヤコビ行列の計算において，変換軸以外の列の値は前ステップの列の値と同じとしていることに問題点がある．

（1） 方向が前ステップと同じとき

1. 初期計算においては，直交軸方向の1つである．その方向で直線上の最小点まで進むことができる．初期の方向決定は，最急降下法によるので，ヤコビ行列のその軸列以外は，0列であったという場合がある．このとき，この点で再び同じ方向が決定される．対策は最小点を求めた時点で，1つの軸方向と同じであったか調べて，そうならばヤコビ行列を軸方向勾配法に

Table 6 Result of improved FLTSQ.

関数		方向決定回数	関数呼出し回数	精度	最終値または時間
①* ENZYME 1	$x_0=(0,0,0,0)$	28 内(凍結 1 回)	94	0.31044×10^{-3}	(0.19295, 0.19267, 0.12512, 13645)
①** BOX	$x_0=(0,10,20)$	13 内(凍結 2 回)	313	0.36403×10^{-11}	(3.1664, 3.1664, 0.51917×10^{-6})
①** 分数関数		46 内(凍結 10 回)	435 (495)	105.62	149.1 秒

①* は，アルゴリズム ① の改良 1，①** は，アルゴリズム ① の改良 2.

よって計算する．改良 1 としてルーチンがつくられている．**10.3** の（1）参照.

2. 途中の計算のとき，前の方向で最小点を見いだしているので前と同じ位置にとどまる．このとき方向決定のための A の計算の分母は 0 になる．この場合，直線探索の前に判定し，同じ方向ならば，ヤコビ行列を軸方向勾配法で計算し方向計算の再計算を行う．これは，プログラムに組込んでない.

（2） 摂動が大きすぎるとき

軸列以外の列に対する軸方向の勾配に関して直線探索の始点と最小点とにおける差が大きくなりやすい．対策は最小点を求めたときに，距離を判定し，ある距離以上ならば，軸方向勾配法によってヤコビ行列を計算する．方向決定の H の計算に，2 つのアルゴリズムが考えられる.

a）方向を軸とする新空間をつくり，前空間で求めた始点，最小点の量を新空間に移して H を計算する.

b）新空間はつくらないで，前空間内で H を計算する．本プログラムは，特に理由はないが b）を用いてある．改良 2 としてルーチンがつくられている．**10.3**（2）参照.

（3） 同じ軸が選択されたとき

方向勾配法によれば，前ステップで選ばれた列は，つづけて再び選ばれないようにしてあるが，ヤコビ行列に誤差があれば，選ばれうる．ヤコビ行列の 1 つの列が 2 回選ばれてその列だけが更新されるので更に誤差の原因になる．対策として，2 つの方法が考えられる.

a）前空間あるいは新空間のいずれかで，ヤコビ行列を軸方向勾配法で計算する.

b）前と同じ列であるならば，次の列を，その列が最後の列ならば最初の列を交換軸列とする.

b）が簡単なのでプログラムは b）を用いて，GR 1 ルーチンに組込む.

10.3 改良したアルゴリズムについて

T-space の更新を行わないことを T-space の凍結とよぶことにする．凍結したときの方向計算は，$H_T{}^0 \leftarrow H_T{}_0$ の代入計算を行い，$T_0{}^{-1}$ の計算，$H_T{}^0 \Leftarrow H_T{}_0$ の計算部をスキップする．直線探索を終り，T-space の更新とヤコビ行列の更新を行う前に，以下の 2 つの場合について判定して処理する.

（1） 1 つの方向に捕縛されるとき

Enzyme 関数について有効であった．方向が軸と一致していたかどうか調べ，一致しているときは，T-space を凍結し，T-space 軸勾配法によってヤコビ行列と勾配を求める．Fig. 5 にフローチャートを示す．ルーチン名は，KAIRYO 1.

（2） 摂動の成分が大きいとき

摂動の成分の内，交換軸以外の大きさがある値 ε1 をこえたとき，T-space を凍結し，T-space 軸勾配法によって，ヤコビ行列と勾配を求める．方向計算の $A=\dfrac{dt\cdot dt'}{dg'\cdot dt}$，$H_T{}^0 \leftarrow H_T{}_0$ 代入計算を行い，方向計算ルーチンの $T_0{}^{-1}$，$H_T{}^0 \Leftarrow H_T{}_0$，$A=\dfrac{\Delta t}{dg_{im}} e_{im}\cdot e_{im}{}'$ の部分をスキップする.

この ε1 は，BOX 関数は 1 であり，分数関数は，0.5 が良い結果であった．Fig. 6 にフローチャートを示す．ルーチン名は，KAIRYO 2.

10.4 改良結果

Table 6 のように，Enzyme, BOX, 分数関数とも，収束した．（ ）内は付録 2 参照.

11. 検 討

テスト関数について，各アルゴリズムの関数呼出し回数の少ない順を **Table 7** のようにまとめなおした.

関数呼出し回数の順を合計すると次のようになる．ただし，不成功の順は 5 とした．①: 17, ③: 9, ④: 29, ⑤: 21. 不成功は除いて平均すると，

①: 2.43, ③: 1.29, ④: 3, ⑤: 2.2 となる.

Table 7 Ascending order table as calling number of function.

アルゴリズム名	関数						
	ROSEN BROCK	3次谷	BEALE	BOX	ENZYME 1	ENZYME 2	分数
① FLTSQ	4	3	3	2	2	1	2
② FLTSQ 1	4	3	3			1	
③ FLTSQ 2	2	1	1	2	1	1	1
④ PWLSQ	3	4	2	5	5	5	5
⑤ NTNGS	1	2	4	1	5	5	3

標準テスト関数の比較結果，オールラウンドで，速いものは，③である．①はオールラウンドであるが，③よりも遅い．2次形式の場合は，①が最も速い．従って，ヤコビ行列が方向勾配法によって良く近似できる関数に対して，本論文で新しく提案した①の方法は，有効であると思う．なお，微分式を与えるかどうかの問題は重要であるので，以下のような検討も合せて行った．DFP 法において，微分式を与えた場合，分数関数について計算した結果 Table 5 の⑥のとおりである．このことについて **9.3** で述べた．この関数においては，微分式を与えた方が良い．微分式を与えたときは，方向決定の精度があがり，くりかえし回数は減少するが，複雑な微分式に対して計算時間は増加する．微分式のない場合のヤコビ行列の計算は，直接探索で得た結果を利用するから計算時間は特別の場合を除いて零である．従って，与えられた問題各々において決定すべきことと思う．

12. おわりに

方程式の優決定系を最小2乗法で解くとき，DFP 法の改良型を提案した．ここで用いられる方向勾配法に関する問題点を指摘し，その対策を合せて示した．直線探索が追加された古典的な Newton-Gauss よりも DFP 法が，ここでとりあげたすべての関数に対して安定しており，しかも，ヤコビ行列が，方向勾配法によって良く近似できるような関数のとき，この改良された方法で試みることは，処理速度を上げる上で，有効であると思う．なお計算機は，MELCON-COSMO 500 を使用した．

参考文献

1) M. J. D. Powell: A Method for Minimizing a Sum of Squares of Non-linear Functions without Calculating Derivatives, Computer J., 7, pp. 303 (1965).
2) R. Fletcher and M. J. D. Powell: A Rapidly Convergent Descent Method for Minimization, Computer J. 6, pp. 163～168 (1963).
3) 松本欣二著：フォートランプログラミング，p. 247，朝倉書店 (1972).
4) J. コワリック，M. R. オズボーン共著，山本善之，小山健夫共訳：非線形最適化問題，p. 165，培風館 (1970).
5) 吉田二朗：有理分数関数の設計について，情報処理，Vol. 19, No. 8, pp. 758～762 (1978).

付録1　記号の説明．

μ, ν, λ, ε：スカラー量．

v_i：ベクトル v の i 成分．

A_{ij}：行列の i 行 j 列成分．

$\|v\|$：v のユークリットノルム，$v=\{v_i\}$，とすると $\|v\|=\sqrt{\sum_i v_i^2}$．

T^{-1}：行列 T の逆行列．

$\kappa_i(A)$：行列 A の i 列ベクトル．

$\rho_i(A)$：行列 A の i 行ベクトル．

e_i：i 成分が1の単位ベクトル．

E：単位行列．

v^t, A^t：それぞれ転置ベクトル，転置行列．

t, x：それぞれ，T-space，×-space の変数．

$g_{t_\times}$：×-space の方向 $t_\times$ の直線上のある点の f の勾配，$df/d\lambda$，ただし λ は直線上の変数．

$g_\times$：×-space のある点の F の勾配．$(\partial F/\partial x)^t$．

$A_\times$：×-space のある点の f の勾配でヤコビ行列．$(\partial f/\partial x)$．

T：方向行列または変換行列．$(\partial x/\partial t)$

$v^t\cdot\omega$：ベクトルの内積，$A^t\cdot B$：行列の積．

$|\exp|$：exp の値の絶対値．

$\max_{i=1,n}(\exp_i)$：$\exp_i$ の値が最大になる i．

$\mathrm{num}_{i=1,n}(\exp_i)$：論理式 $\exp_i$ が真になる回数．

$A\leftarrow B$：A の値を B の値と等しくする．または，B を A に代入する．

$A\Leftarrow B$：B の値を用いて，ある計算をして，その結果を A の値とする．すなわち，B から A を導出する．ここで A は変数，B は式，変数，または定数．

$g_\times^0$, T_0：右肩，右下に0が添えてある記号は，前ステップの量．

小英字は，特に説明がないときベクトル．

大英字は，特に説明がないとき行列．

ギリシャ文字は，特に説明がないときスカラー定数．

特に説明がなければ，数学上の記号と同じ意味である．

付録2

勾配計算時の関数の呼出し回数．

(1)　軸勾配法．

直線探索で求められた $f(x)$ を用いる．x の周りで f_p を n 回計算する．$\therefore$ n・IC 回．

(2)　方向勾配法

直線探索で求められた f, f_p を用いる．$\therefore$ 0回．もし，Fig. 3 で，$|\lambda|\geqq 10^{-9}$ が NO のときが K 回あ

るとすると，$n \cdot K$ 回.

（3） FLTSQ の改良法

凍結回数を K 回とすると K 回，軸勾配法が用いられる．$\therefore$ $n \cdot K$ 回

ここで，IC は方向決定回数.

この方法を考慮した関数呼び出し回数を分数関数の ③ と ① に（ ）で記入してある．Table 5，Table 6 参照.

（昭和52年6月28日受付）
（昭和53年4月17日再受付）