

数理計画法の計算方法による研究 (I)

Gradient Projection Method の Programming について

片 山 剛 之 丞

Study on Calculation Method of Mathematics Programming (1)

By Gonojo KATAYAMA

Automatic design is just going to advance to optimal design and this is the last target to take use of computer. The design is a process to get the best combination of design parameter; is a system engineering. This is equal to get the summit of objective function in plural parameter space. In the optimal programming at management, linear programming and dynamic programming is much used. Modified function and objective function is generally complicated nonlinear function, so the design calculation calculate on nonlinear programming. In this report, using of the gradient projection method, indicate the program of the gradient projection method in order to solve the nonlinear problem.

〔1〕 ま え が き

現在、自動設計は最適設計に進みつつあり、これは計算機利用の最終的な目標である。設計とは設計パラメータの最もよい組合せを求める過程であり、システムエンジニアリングである。経営面における最適計画に線形計画法やダイナミック・プログラミングがよく使用されるが、設計計算は制限関数も目的関数も一般に複雑な非線形計画法によらねばならない。

非線形計画法は制約が不等式であり、非常に多くの変数をもつ大きなシステムに用いられる。変数は等式によって与えられる場合もある。この目的は制約および等式を満足する変数で、ある関数を最大にすることである。非線形計画法は制約および等式が線形であり目的関数のみが非線形の場合と、制約・等式および目的関数が非線形の場合がある。まず、制約および等式が線形で目的関数のみが非線形の場合について、勾配射影法を使用して、非線形問題を解くために、勾配射

影法のプログラムを示した。

〔2〕 アルゴリズム

m 個の変数 $x_i (i=1, \dots, m)$ を考え、これはユークリッド空間の点を表わし、 $\mathbf{x} = \{x_1, x_2, \dots, x_m\}$ とする。 \mathbf{x}^T により、ベクトル \mathbf{x} の転置ベクトルを表わす。 m 次元ベクトル \mathbf{x} の関数 $F(\mathbf{x})$ があり $F(\mathbf{x})$ を最大ならしめる点を求めるものとする。

制約式は $\sum_{j=1}^m n_{ij}x_j - b_i \geq 0; (i=1, \dots, k)$ の形をしている。ただし、 n_{ij} は $\sum_{j=1}^m (n_{ij})^2 = 1; (i=1, \dots, k)$ のように標準化されている。制約式はユークリッド m 次元空間 (E_m) において、有界な凸領域 (R) を形成する。 k 個の制約式をそれぞれベクトル \mathbf{n}_i に対応させ、 $\mathbf{n}_i = \{n_{i1}, n_{i2}, \dots, n_{im}\}; (i=1, \dots, k)$ とする。制約式は $\mathbf{x}^T \mathbf{n}_i - b_i = \lambda_i(\mathbf{x}) \geq 0$ と表わされ、 $\lambda_i(\mathbf{x}) = 0$ で定義される $(m-1)$ 次元多面体は H_i によって定義される超平面である。

$m \times k$ 行列を \mathbf{N}_k , k 次元ベクトルを \mathbf{b}_k とすれば、

制約式は $\mathbf{N}_k^T \mathbf{x} - \mathbf{b}_k \geq 0$ となる。 $m \times m$ 正方形行列 $\tilde{\mathbf{p}}_q$ を $\tilde{\mathbf{p}}_q = \mathbf{N}_q (\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q^T$ とすると、行列 $\tilde{\mathbf{p}}_q$ は E_m 内の任意のベクトルを \tilde{Q} に入れる射影行列である。 $m \times m$ 行列 \mathbf{p}_q を $\mathbf{p}_q = \mathbf{I} - \tilde{\mathbf{p}}_q = \mathbf{I} - \mathbf{N}_q (\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q^T$ とすると、 \mathbf{p}_q は E_m 内の任意のベクトルを交り (Q) に射影する射影行列である。 目的関数を $F(\mathbf{x})$ とすると、 $F(\mathbf{x}) = F(x_1, x_2, \dots, x_m)$ は R において定義され、連続で、 x_i に関して有界な 2 階偏導関数が存在すると仮定し、 $F(\mathbf{x})$ は R 内で凹関数であると仮定する。

制約のない凹関数に対して、 \mathbf{x}^0 が最大点となるための必要十分条件は、 $\mathbf{g}(\mathbf{x}^0) = 0$ となることである。ここで、 $\mathbf{g}(\mathbf{x})$ は $F(\mathbf{x})$ の勾配である。もし、このような点が領域 R の内部で求まるならば、それは R 内の全域的最大であり、 $F(\mathbf{x})$ の内部全域的最大であり、もし R の内部において $\mathbf{g}(\mathbf{x}^0) \neq 0$ ならば、その時は全域的最大は境界上にあり、制約された全域的最大となる。 \mathbf{x}^0 は正確に q 個 ($1 \leq q \leq m$) の超平面上にある R の境界上の点とする。この q 個の超平面は一次独立であると仮定する。これらの超平面の交りを多様体 Q とすると、点 \mathbf{x}^0 が $F(\mathbf{x})$ の制約された全域的最大となるための必要十分条件は $\mathbf{p}_q \mathbf{g}(\mathbf{x}^0) = 0$ および $(\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q \mathbf{g}(\mathbf{x}^0) \leq 0$ である。

全域的最大でない任意の実行可能解 \mathbf{x}_v が与えられると、 $F(\mathbf{x})$ のより大きな値をもつ別の実行可能な点を見つけるアルゴリズムが得られる。 \mathbf{x}_v は q 個の一次独立な超平面の交り Q 上に存在すると仮定する。 $q=0$ は $\mathbf{p}_q = \mathbf{I}$, かつ $\mathbf{p}_q \mathbf{g} = \mathbf{g}$ であるとする。まず、 $\mathbf{g}_v = \mathbf{g}(\mathbf{x}_v)$ および $\mathbf{p}_q \mathbf{g}_v$ を計算する。もし、 $\mathbf{p}_q \mathbf{g}_v = 0$ かつ $\mathbf{r} \leq 0$ ($\mathbf{r} = (\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q^T \mathbf{g}_v$) ならば、 \mathbf{x}_v は全域的最大である。もし、 $\|\mathbf{p}_q \mathbf{g}_v\| > \max\{0, 1/2 r_q r_v^{-1/2}\}$ ならば、 $\mathbf{z} = \mathbf{p}_q \mathbf{g}_v / \|\mathbf{p}_q \mathbf{g}_v\|$ より \mathbf{z} を計算する。また、 $\|\mathbf{p}_q \mathbf{g}_v\| < 1/2 r_q r_v^{-1/2}$ ならば、 Q から H_q を削除し、 \mathbf{N}_{q-1} と対応する \mathbf{p}_{q-1} を計算し、 $\mathbf{z} = \mathbf{p}_{q-1} \mathbf{g} / \|\mathbf{p}_{q-1} \mathbf{g}\|$ より \mathbf{z} を計算する。ただし、 $r_g = \max\{r_i\} > 0, r_v \geq \|(\mathbf{N}_q^T \mathbf{N}_q)^{-1}\|$ とする。 $\tau_i = -\lambda_i(\mathbf{x}^0) / \mathbf{z}^T \mathbf{n}_i$; ($i=q+1, \dots, k$) より $\tau_m = \min\{\tau_i \geq \delta\}$ を計算し、 $\mathbf{x}'_{v+1} = \mathbf{x}_v + \tau_m \mathbf{z}$ とする。 $\mathbf{g}'_{v+1} = \mathbf{g}(\mathbf{x}'_{v+1})$ を計算する。 $\mathbf{z}^T \mathbf{g}'_{v+1} \geq 0$ ならば $\mathbf{x}_{v+1} = \mathbf{x}'_{v+1}$ とし、 $\tau_m = \min\{\tau_i \geq \delta\}$ における最小値 τ_i に対応する H_i を Q に加える。また、 $\mathbf{z}^T \mathbf{g}'_{v+1} < 0$ ならば、 $\rho = \mathbf{z}^T \mathbf{g}_v / (\mathbf{z}^T \mathbf{g}_v - \mathbf{z}^T \mathbf{g}'_{v+1})$, $\mathbf{x}_{v+1} = \rho \mathbf{x}'_{v+1} + (1-\rho) \mathbf{x}_v$ とする。交り Q は変化せずそのままである。

各ステップにおいて、交り Q は変化しなくてもよ

い。ある超平面が落されるか、または付け加えられるかもしれない。あるいは、一つの超平面が落され、他の超平面が付け加えられるかもしれない。

[3] プログラムで用いる記号

3-1 変数の説明

X(i), X1(i)	変数 $i=1, \dots, m$
F	目的関数
UK(i, j)	制約式の係数 $\mathbf{N}_k, j=1, \dots, k$
B(i)	制約式の定数項 $i=1, \dots, k$
CONSTF(i)	制約式の値 $i=1, \dots, k$
GRADIE(i)	目的関数の勾配 $\nabla f, i=1, \dots, m$
V(i, j)	逆行列 $(\mathbf{N}_q^T \mathbf{N}_q)^{-1}, i=j=1, \dots, m$
PQG(i)	$\mathbf{p}_q \mathbf{g} = \mathbf{g} - \mathbf{N}_q (\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q^T \mathbf{g}$
RR(i)	$\mathbf{r}_{q-1} = (\mathbf{N}_{q-1}^T \mathbf{N}_{q-1})^{-1} \mathbf{N}_{q-1}^T \mathbf{g}$
R(i)	$\mathbf{r}_q = (\mathbf{N}_q^T \mathbf{N}_q)^{-1} \mathbf{N}_q^T \mathbf{g}$
RN(i)	$(\mathbf{N}_{q-1}^T \mathbf{n}_q)$
R1(i)	$\mathbf{N}_q^T \mathbf{g}$
PQN(i)	$\mathbf{p}_{q-1} \mathbf{n}_q$
Z(i)	$\mathbf{z} = \mathbf{p}_q \mathbf{g} / \ \mathbf{p}_q \mathbf{g}\ $
MJU(i)	一次従属
M1(i)	正の制約式番号
M2(i)	零の制約式番号
M3(i)	負の制約式番号
AO	$\ \mathbf{p}_{q-1} \mathbf{n}_q\ $
PQGNOR	$\ \mathbf{p}_q \mathbf{g}\ $
TAU	τ_i
ALFA	$\ \mathbf{V}_q\ ^{1/2}$
XX	$z(x_i)g(x_i)$
YY	$z(x_i)g(x_{i+1})$
M, MKM	変数の個数
K, KMK	制約式の個数
K1	正の制約式の個数
K2	零の制約式の個数
K3	負の制約式の個数
LL	超平面を落とすときの制約式番号
IP	一次従属の個数
ERROR	誤差
IIP	一次従属のうち、 z に対して負となるもののカウント
IPIP	一次従属のうち、 z に対して負となるものの制約式の番号
KQA	一次独立の個数
KQ	逆行列に加える個数
MK1	$M+1$
KM1	$K+1$
ISTEP	ステップの番号

3-2 サブルーチンの説明

GRASUB	目的関数の値と ∇f の値を求める。
CONSUB	制約式の値を求める。
VQSUB	逆行列を作る。
VQLSUB	超平面を一つ削除する。
RSUB	$r_q, p_q, \ p_q\ $ を計算する。
INDSUB	一次従属の処理をする。
IOCSUB	一次従属の時の制約式の値を調べる。
REGSUB	初期値が領域外の時、 x_i を領域内に入れる。

[4] プログラム

```

1      COMMON UK(30,60), V(30,30), X(30), X1(30), M1(60), M2(60), M3(60),
1      PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2      B(60), Z(60), CONSTF(60), MJU(60)
2      COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1      STAND, MKM, KMK, MK1, KM1, IIP, IPIP
3      COMMON ALP
C     IN PUT
4      DIMENSION XA(30), XB(30)
5      READ(5,111) M, K, ERROR
6      READ(5,112) ((UK(I,J), I=1, M), J=1, K)
7      READ(5,112) (B(I), I=1, K)
8      READ(5,112) (X(I), I=1, M)
9      READ(5,140) ALP
10     ISTEP=0
C     UK(I,J), B(I) IS STANDARD
11     260 DO 210 I=1, K
12         STAND=0.0
13         DO 230 J=1, M
14             230 STAND=STAND+UK(J,I)**2
15             STAND=SQRT(STAND)
16             DO 250 JJ=1, M
17                 250 UK(JJ,I)=UK(JJ,I)/STAND
18                 B(I)=B(I)/STAND
19     210 CONTINUE
20     CALL CONSUB
21     K1=0
22     K2=0
23     K3=0
24     MKM=M
25     KMK=K
26     IP=0

```

```

27      DO 270 I=1, K
28      IF(CONSTF(I).GT.ERROR) GO TO 290
29      IF(CONSTF(I).LT.-ERROR) GO TO 310
30      K2=K2+1
31      M2(K2)=I
32      GO TO 270
33      290 K1=K1+1
34      M1(K1)=I
35      GO TO 270
36      310 K3=K3+1
37      M3(K3)=I
38      270 CONTINUE
39      IF(K3.EQ.0) GO TO 850
40      CALL REGSUB
41      850 IF(K2.EQ.0) GO TO 330
C      ADDING TO INVERS
42      650 KQA=1
43      V(1,1)=1.0
44      IF(K2.EQ.1) GO TO 730
45      KQ=K2-1
46      K2=1
47      630 CALL VQSUB
48      730 CALL GRASUB
49      CALL RSUB
C      MAX R(I) O MOTOMERU
50      LL=0
51      BETA=ERROR
52      DO 370 I=1, K2
53      IF(BETA.GT.R(I)) GO TO 370
54      LL=I
55      BETA=R(I)
56      370 CONTINUE
57      ISTEP=ISTEP+1
C      OUT PUT
58      WRITE(6, 113) ISTEP
59      DO 830 I=1, M
60      830 WRITE(6, 115) I, X(I)
61      WRITE(6, 116) F
C      JUDGING OPTIMAL SOLUTION
62      IF(LL.NE.0) GO TO 390
63      IF(PQGNOR.GE.ERROR) GO TO 390
64      770 IF(K3.EQ.0) GO TO 220
65      IF(ABS(F).GE.ERROR) GO TO 240
66      WRITE(6, 117)

```

```

67      M=MKM
68      K=KMK
69      GO TO 260
70      240 WRITE(6,119)
71      STOP
72      220 WRITE(6,114)
73      999 STOP
74      390 ALFA=0.0
75      DO 410 I=1,K2
76      DO 410 J=1,K2
77      410 ALFA=ALFA+V(I,J)**2
78      ALFA=SQRT(ALFA)
79      ALFA=SQRT(ALFA)
80      PG=PQGNOR
81      BA=BETA/(2.*ALFA)
82      IF(PG.GT.BA) GO TO 430
C      DROPPING FROM INVERS
83      CALL VQLSUB
84      CALL RSUB
C      Z(I) NO KEISAN
85      430 DO 450 I=1,M
86      450 Z(I)=PQG(I)/PQGNOR
87      IF(IP.EQ.0) GO TO 720
88      IF(PG.GT.BA) GO TO 720
89      CALL INDSUB
90      IF(IIP.EQ.0) GO TO 720
91      MJU(IIP)=MJU(IP)
92      IP=IP-1
93      K2=K2+1
94      M2(K2)=M2(1)
95      M2(1)=IPIP
96      GO TO 650
C      TAU NO KEISAN
97      720 TAU=100000000.0
98      DO 470 I=1,K1
99      II=M1(I)
100     XVALUE=0.0
101     YVALUE=0.0
102     DO 490 J=1,M
103     XVALUE=XVALUE+UK(J,II)*X(J)
104     490 YVALUE=YVALUE+UK(J,II)*Z(J)
105     XVALUE=XVALUE-B(II)
106     VALUEY=ABS(YVALUE)
107     IF(VALUEY.LE.ERROR) GO TO 470

```

```

108      EVALUE=-XVALUE/YVALUE
109      IF(EVALUE.LE.ERROR) GO TO 470
110      IF(EVALUE.GT.TAU) GO TO 470
111      TAU=EVALUE
112      470 CONTINUE
113      YY=0.0
114      DO 510 I=1, M
115      510 YY=YY+Z(I)*GRADIE(I)
116      DO 530 I=1, M
117      X1(I)=X(I)
118      530 X(I)=X(I)+TAU*Z(I)
119      CALL GRASUB
120      XX=0.0
121      DO 550 I=1, M
122      550 XX=XX+Z(I)*GRADIE(I)
123      IF(XX.LT.-ERROR) GO TO 570
124      CALL CONSUB
125      K11=0
126      K4=K2
127      DO 590 I=1, K1
128      J=M1(I)
129      IF(ABS(CONSTF(J)).LE.ERROR) GO TO 610
130      K11=K11+1
131      M1(K11)=J
132      GO TO 590
133      610 K4=K4+1
134      M2(K4)=J
135      590 CONTINUE
136      K1=K11
137      KQ=K4-K2
138      IF(IP.EQ.0) GO TO 320
139      CALL IOCSUB
140      320 IF(K2.NE.0) GO TO 630
141      K2=K4
142      GO TO 650
143      570 CALL GRASUB
144      ISTEP=ISTEP+1
C      OUT PUT
145      WRITE(6,113) ISTEP
146      DO 690 I=1, M
147      690 WRITE(6,115) I, X(I)
148      WRITE(6,116) F
C      TAU NO HOKAN
149      TAU=YY/(YY-XX)

```

```

150      DO 670 I=1, M
151      670 X(I)=TAU*X(I)+(1.0-TAU)*X1(I)
152      IF(IP.EQ.0) GO TO 340
153      CALL CONSUB
154      CALL IOCSUB
155      340 IF(K2.GT.0) GO TO 730
156      330 CALL GRASUB
157      IP=0
158      ISTEP=ISTEP+1
C      OUT PUT
159      WRITE(6,113) ISTEP
160      DO 810 I=1, M
161      810 WRITE(6,115) I, X(I)
162      WRITE(6,116) F
163      PQGNOR=0.0
164      DO 750 I=1, M
165      PQG(I)=GRADIE(I)
166      750 PQGNOR=PQGNOR+PQG(I)**2
167      PQGNOR=SQRT(PQGNOR)
168      IF(PQGNOR-ERROR) 770,770,430
C      FORMAT
169      111 FORMAT(2I5,F15.10)
170      112 FORMAT(4F15.10)
171      140 FORMAT(F20.11)
172      113 FORMAT(1H1,20X,27HSOLUTION OF GRADIENT METHOD,///,35X,
173      1      4HSTEP,I7,////)
174      114 FORMAT(////,20X,16HOPTIMAL SOLUTION)
175      115 FORMAT(/,15X,2HX(,I3,2H) ,E20.11)
176      116 FORMAT(///,15X,7HF( X ) ,E20.11)
177      117 FORMAT(/,20X,27HTHIS POINT IS IN THE REGION)
178      119 FORMAT(/,20X,26HTHERE IS NO FEASIBLE POINT)
179      END

1      SUBROUTINE VQSUB
2      COMMON UK(30,60),V(30,30),X(30),X1(30),M1(60),M2(60),M3(60),
3      1      PQG(30),GRADIE(30),RN(30),RR(30),PQN(30),R1(30),R(30),
4      2      B(60),Z(60),CONSTF(60),MJU(60)
5      COMMON M,K,K1,K2,K3,KQA,KQ,AO,LL,TAU,PQGNOR,ERROR,F,IP,
6      1      STAND,MKM,KMK,MK1,KM1,IIP,IPIP
7      COMMON ALP
8      DO 500 IJ=1,KQ
9      KQA=KQA+1

```

```

7      NM=M2(KQA)
8      DO 1010 I=1, K2
9      RN(I)=0.0
10     II=M2(I)
11     DO 1020 J=1, M
12     RN(I)=RN(I)+UK(J, II)*UK(J, NM)
13 1020 CONTINUE
14 1010 CONTINUE
15     DO 1040 I=1, K2
16     RR(I)=0.0
17     DO 1040 J=1, K2
18     RR(I)=RR(I)+V(I, J)*RN(J)
19 1040 CONTINUE
20     AO=0.0
21     DO 1050 I=1, M
22     PQN(I)=0.0
23     DO 1060 J=1, K2
24     JJ=M2(J)
25     PQN(I)=PQN(I)+UK(I, JJ)*RR(J)
26 1060 CONTINUE
27     PQN(I)=UK(I, NM)-PQN(I)
28     AO=AO+PQN(I)**2
29 1050 CONTINUE
30     IF(ABS(AO).GT.ERROR) GO TO 1090
31     IP=IP+1
32     MJU(IP)=NM
33     GO TO 500
34 1090 M2(K2+1)=NM
35     DO 2030 I=1, K2
36     DO 2030 J=I, K2
37     AB=RR(I)*RR(J)/AO
38     V(I, J)=V(I, J)+AB
39     V(J, I)=V(I, J)
40 2030 CONTINUE
41     II=K2+1
42     DO 2050 I=1, K2
43     V(I, II)=-RR(I)/AO
44     V(II, I)=V(I, II)
45 2050 CONTINUE
46     V(II, II)=1./AO
47     K2=K2+1
48 500 CONTINUE
49     KQA=K2
50     RETURN

```


51 END

```
1           SUBROUTINE VQLSUB
2           COMMON UK(30,60),V(30,30),X(30),X1(30),M1(60),M2(60),M3(60),
1           PQG(30),GRADIE(30),RN(30),RR(30),PQN(30),R1(30),R(30),
2           B(60),Z(60),CONSTF(60),MJU(60)
3           COMMON M,K,K1,K2,K3,KQA,KQ,AO,LL,TAU,PQGNOR,ERROR,F,IP,
1           STAND,MKM,KMK,MK1,KM1,IIP,IPIP
4           COMMON ALP
5           DO 1100 I=1,K2
6           W=V(I,LL)
7           V(I,LL)=V(I,K2)
8           V(I,K2)=W
9           1100 CONTINUE
10           DO 1111 J=1,K2
11           W=V(LL,J)
12           V(LL,J)=V(K2,J)
13           V(K2,J)=W
14           1111 CONTINUE
15           MA=M2(LL)
16           M2(LL)=M2(K2)
17           K1=K1+1
18           M1(K1)=MA
19           KQ=K2-1
20           IF(KQ.EQ.0) GO TO 1045
21           DO 1043 I=1,KQ
22           DO 1043 J=I,KQ
23           ABC=V(I,K2)*V(K2,J)/V(K2,K2)
24           V(I,J)=V(I,J)-ABC
25           V(J,I)=V(I,J)
26           1043 CONTINUE
27           GO TO 1046
28           1045 V(1,1)=0.
29           1046 K2=K2-1
30           KQA=KQA-1
31           RETURN
32           END
```

```
1           SUBROUTINE IOCSUB
2           COMMON UK(30,60),V(30,30),X(30),X1(30),M1(60),M2(60),M3(60),
```

```

1          PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2          B(60), Z(60), CONSTF(60), MJU(60)
3  COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1          STAND, MKM, KMK, MK1, KM1, IIP, IPIP
4  COMMON ALP
5  IPP=0
6  DO/7100 I=1, IP
7  J= MJU(I)
8  IF(ABS(CONSTF(J)).LE.ERROR) GO TO 7110
9  K1=K1+1
10 M1(K1)=J
11 GO TO 7100
12 7110 IPP=IPP+1
13     MJU(IPP)=J
14 7100 CONTINUE
15     IP=IPP
16     RETURN
17     END

```

```

1  SUBROUTINE INDSUB
2  COMMON UK(30, 60), V(30, 30), X(30), X1(30), M1(60), M2(60), M3(60),
1          PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2          B(60), Z(60), CONSTF(60), MJU(60)
3  COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1          STAND, MKM, KMK, MK1, KM1, IIP, IPIP
4  COMMON ALP
5  SHIMIN= -ERROR
6  IIP=0
7  IPIP=0
8  DO 6100 I=1, IP
9  J= MJU(I)
10 SHITA=0.0
11 DO 6110 II=1, M
12 6110 SHITA=SHITA+UK(II, J)*Z(II)
13     IF(SHITA.GE.SHIMIN) GO TO 6100
14     IIP=I
15     IPIP=J
16     SHIMIN=SHITA
17 6100 CONTINUE
18     RETURN
19     END

```

```

1      SUBROUTINE REGSUB
2      COMMON UK(30,60), V(30,30), X(30), X1(30), M1(60), M2(60), M3(60),
1          PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2          B(60), Z(60), CONSTF(60), MJU(60)
3      COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1          STAND, MKM, KMK, MK1, KM1, IIP, IPIP
4      COMMON ALP
5      MK1=M+1
6      MK3=M+K3
7      KM1=K+1
8      KM3=K+K3
9      DO 5100 I=MK1, MK3
10     DO 5100 J=1, K
11     5100 UK(I, J)=0.0
12     DO 5110 I=1, MK3
13     DO 5110 J=KM1, KM3
14     5110 UK(I, J)=0.0
15     DO 5120 I=1, K3
16     J=M3(I)
17     M=M+1
18     K=K+1
19     K1=K1+1
20     K2=K2+1
21     M1(K1)=K
22     M2(K2)=J
23     UK(M, J)=1.0
24     UK(M, K)=1.0
25     B(K)=0.0
26     5120 X(M)=-CONSTF(J)
C     UK(I, J), B(I) IS STANDARD
27     DO 5130 I=1, K3
28     JJ=M3(I)
29     STAND=0.0
30     DO 5140 J=1, M
31     5140 STAND=STAND+UK(J, JJ)**2
32     STAND=SQRT(STAND)
33     DO 5150 II=1, M
34     5150 UK(II, JJ)=UK(II, JJ)/STAND
35     5130 B(JJ)=B(JJ)/STAND
36     RETURN
37     END

```

```

1      SUBROUTINE CONSUB
2      COMMON UK(30,60), V(30,30), X(30)X, 1(30), M1(60), M2(60), M3(60),
1          PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2          B(60), Z(60), CONSTF(60), MJU(60)
3      COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1          STAND, MKM, KMK, MK1, KM1, IIP, IPIP
4      COMMON ALP
5      DO 1200 J=1, K
6      CONSTF(J)=0.0
7      DO 1210 I=1, M
8      CONSTF(J)=CONSTF(J)+UK(I, J)*X(I)
9      1210 CONTINUE
10     CONSTF(J)=CONSTF(J)-B(J)
11     1200 CONTINUE
12     RETURN
13     END

```

```

1      SUBROUTINE RSUB
2      COMMON UK(30,60), V(30,30), X(30), X1(30), M1(60), M2(60), M3(60),
1          PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30),
2          B(60), Z(60), CONSTF(60), MJU(60)
3      COMMON M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP,
1          STAND, MKM, KMK, MK1, KM1, IIP, IPIP
4      COMMON ALP
5      DO 2700 I=1, K2
6      R1(I)=0.0
7      II=M2(I)
8      DO 2700 J=1, M
9      R1(I)=R1(I)+GRADIE(J)*UK(J, II)
10     2700 CONTINUE
11     DO 2900 I=1, K2
12     R(I)=0.0
13     DO 2900 J=1, K2
14     R(I)=R(I)+V(I, J)*R1(J)
15     2900 CONTINUE
16     PQGNOR=0.0
17     DO 2300 I=1, M
18     PQG(I)=0.0
19     DO 2350 J=1, K2
20     JJ=M2(J)
21     PQG(I)=PQG(I)+UK(I, JJ)*R(J)

```

```

22      2350 CONTINUE
23          PQG(I)=GRADIE(I)-PQG(I)
24          PQGNOR=PQGNOR+PQG(I)**2
25      2300 CONTINUE
26          PQGNOR=SQRT(PQGNOR)
27          RETURN
28          END

```

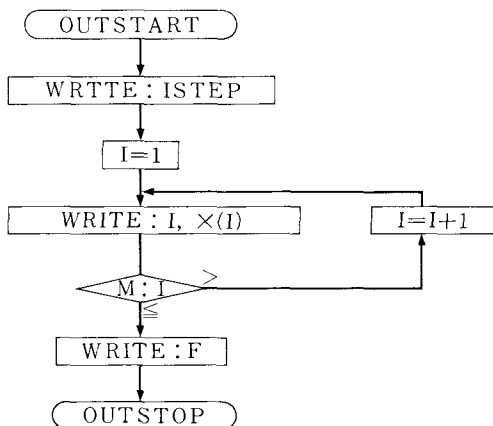
```

1          SUBROUTINE GRASUB
2          COMMON UK(30,60),V(30,30),X(30),X1(30),M1(60),M2(60),M3(60),
1          PQG(30),GRADIE(30),RN(30),RR(30),PQN(30),R1(30),R(30),
2          B(60),Z(60),CONSTF(60),MJU(60)
3          COMMON M,K,K1,K2,K3,KQA,KQ,AO,LL,TAU,PQGNOR,ERROR,F,IP,
1          STAND,MKM,KMK,MK1,KM1,IIP,IPIP
4          COMMON ALP
5          IF(K3.EQ.0) GO TO 5111
6          DO 5200 I=1,MKM
7      5200 GRADIE(I)=0.0
8          F=0.0
9          DO 5210 I=MK1,M
10         GRADIE(I)=-1.0
11      5210 F=F-X(I)
12         GO TO 5500
13      5111 CONTINUE
14         F=F(X)
15         GRADIE(1)=∇1F(X)
           ⋮
           ⋮
         GRADIE(I)=∇1F(X)
           ⋮
           ⋮
         GRADIE(M)=∇MF(X)
5500 RETURN
        END

```

[5] あとがき

INPUT の形式は、制約式は $Ax - b \geq 0$ の形に直して入れなければならない。M, K は 2I5, ERROR は F15.10 とし、UK(I, J), B(I), X(I) は 4F15.10 とする。次に、OUTPUT の形式についてフローチャートを次に示す。



また、目的関数は問題によって全て変化するから、GRASUB の F と GRADIE (I) は目的関数に適した式を代入しなければならない。

COMMON-DIMENSION は、

UK(30, 60), V(30, 30), X(30), X1(30), M1(60), M2(60), M3(60), PQG(30), GRADIE(30), RN(30), RR(30), PQN(30), R1(30), R(30), B(60), Z(60), CONSTF(60), MJU(60), M, K, K1, K2, K3, KQA, KQ, AO, LL, TAU, PQGNOR, ERROR, F, IP, STAND, MKM, KMK, MK1, KM1, IIP, IPIP

である。

[6] 参考文献

- 1) J. B. Rosen; "The Gradient Projection Method For Nonlinear Programming." Part I J. Soc. Indust. Appl. Mathematics. Vol. 8. No. 1. 1960
- 2) J. B. Rosen; "The Gradient Projection Method For Nonlinear Programming." Part II J. Soc. Indust. Appl. Mathematics. Vol. 9. No. 4. 1961
- 3) 坂和爱幸; "最適制御過程Ⅱ" 共立出版
- 4) 三根久; "オペレーションズ・リサーチ (上)" 朝倉書店刊
- 5) マコーミク: サルバドリ; "数値計算プログラム" サイエンス社刊
- 6) 藤野, 機械学会誌, 73-612 (昭45-1), 76