UDC 518.5；621-50+621.503.5

フィードバック形計算方式と最適化計算

菊　池　豊　彦*
Toyohiko Kikuchi

Feed Back Type Calculation and Optimization Techniques

Synopsis

The content of this paper is divided into two parts.

The first part describes the new concept, Feed Back Type Calculation Formula, which was developed by our group, and explains the engineering techniques, which were also formulated by our group, to be applied to the optimization problems.

The second part shows the new optimization technique, Modified Direct Search Method, developed by our group, and exhibits the outline of the Linear Programming Code for FACOM 222.

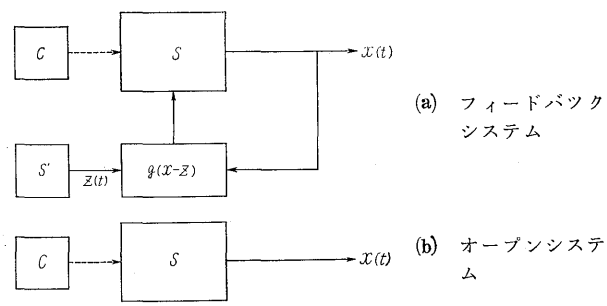
I. まえがき

計算機の活用と発展，および応用数学の進歩は，従来の計算方式より，さらに進んだ方式をわれわれに要求する．そこで，まず「フィードバック形計算方式」なる概念を開発し，計算方式として，あらゆる種類の問題に対して共通な場をもたせるとともに，数学モデルの作成と修正に関する3種類のエンジニヤリング的なアルゴリズムを開発および確立した．これらを記述し，さらにこれらの作業の副産物として開発された「修正直接法」，当社にある電子計算機 FACOM 222 のために作成した線形計画法コード (L•P 25) の特長を紹介する。

II. フィードバック形計算方式とは

第1図は自動制御に関する教科書ならば，必ず記述されているものである．すなわち図中，S は実際の物理系であり，$x(t)$ がその状態を表示するもので，また S' は他の実際の，あるいは仮想な物理系でその状態は $z(t)$ で表示されている．これらの2種の状態変数 $x(t)$ と $z(t)$ がなんらかの方法で比較され，その結果，物理系 S に $g(x-z)$ の形で影響がおよぼされる．これがフィードバック制御といわれ，第1図(a)に示したものになる．ところで，この他に第1図(b)に示したようなものも存在し，これはオープン制御ともいわれている。

いま，計算方式の立場からこの図をながめてみよう．この場合には，系 S は物理系を表現している方程式群をあらわし，$x(t)$ は計算結果である．ここで計算に使用される入力はすべて C から与えられている．このように考えるとわれわれが日常計算している方式は，その大半が第1図(b)に相当している．すなわち，ある式群 S に入力 C を与え計算を遂行し，計算結果 $x(t)$ をそのままとりだす．このような方式をとらざるをえなかったのは，数値解を求めるといった単純な操作を人力でおこなっていたことと，同図(a)のような計算方式を用いたくても，そのための研究が充分にされていなかったことによると考えられる．しかし，電子計算機の発展とその活用および応用数学の進歩が前述のオープン形計算方式より，同図(a)で示されるフィードバック形計算方式をわれわれに要求する．すなわちフィードバック形計算方式はフィードバック制御の思想に基づいた計算方式である．さらに詳細に説明してみよう。



(a) フィードバックシステム

(b) オープンシステム

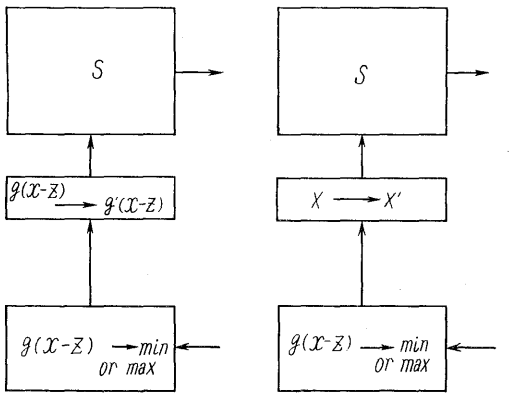
第1図　2種類の異なった技術的手法の概念図
Fig. 1. Two different engineering techniques

われわれは C から入力をえて S なる方程式群を解くことにより，$x(t)$ なる計算結果を得る．他方，物理系 S の最適な状態を得るために，他の式群 S' より，最適にすべき変数群 $z(t)$ が与えられ，その結果，計算結果 $x(t)$ と変数群 $z(t)$ が比較され，$g(x-z)$ の形で S に影響をおよぼす．すなわちもう一度計算を行なう．このループは，$g(x-z)$ が最小あるいは最大になるまで生きることになる．すなわち単に物理系，S を表現している式群を解くのみでなく，それらの式群を満足するととも

* 開発部

に，その系が最適であるような結果を求めることが，フィードバック形計算方式である．この考え方は，その計算対象に関係なくあらゆる問題に共通した概念である．このようにしてフィードバック計算方式を定義すると，最適化計算と呼ばれるものが非常に明白になる。

第２図にフィードバック計算方式の主部分のブロックダイアグラムを記述した．俗にいわれる最適化計算は，第１図(a)の $g(x-z)$ の部分または S の部分に入るべきであるが説明上，第２図では，S と $g(x-z)$ の間に入れてある。



(a) 数学モデルの修正　　(b) 最適化手法

第２図　フィードバック計算方式の主要部分

Fig. 2. Main part of feed back calculation system

ところで最適化計算は，次の２種類にわけられる。

1) 最適化手法，計算結果 $x(t)$ を変化して $x'(t)$ にする．この変化すべき手順を示したのが最適化手法といわれるものであり，第２図(b)にダイアグラムを示す。

2) モデル修正，第２図(a)に示したごとく，$g(x-z)$，そのものを変化させて $g'(x-z)$ にし，最適化をはかる。

いずれにしても，この種の計算に用いられる数学は共通したものであり，どの解析方法がどちらにより良いかといった判別は不可能である．要は，S と $g(x-z)$ を充分に考察することによってのみ，使用すべき最適な手法が定められることを認識する必要がある。

なお，第１図において計算結果 x を特に時間の関数として $x(t)$ と表示したのは次の理由からである。

われわれが日常行なっている計算では，時間が陽に表示されている場合と，されていない場合がある．しかしわれわれが興味を持っている物理系は，最も基本的な独立変数が時間であることを考えると，どちらにおいても時間の関数として表示し，陽に表示されていない場合にはある定めた時間での解であると考えるべきである．すなわちわれわれは数学によって物理系をコピーしているのにすぎないのであるから，物理系がその時間的尺度に関係なく，時間とともに必ず変化することからして，われわれの得た静的な解もすべて動的な解であると解釈しよう．そうすればフィードバック計算方式がそのまま計算機制御，経営分析などの方式としても使用できることになる．これは，常微分方程式の解を，その初期状態からの変化であると考えることによってダイナミックプログラミングが生まれたことと同一視できないであろうか。

III. 非線形な数学モデルの作成と修正に対する考察

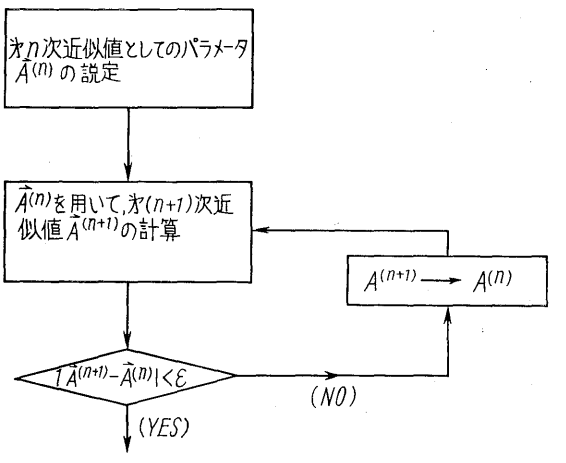
われわれが物理系から得られるものは，その物理系を表現する方程式群でなく，われわれが勝手に定めたその物理系を構成している独立変数群に対する一連のデータである．この入手したデータをもとにして逆に物理系を表現する数学モデルを作成することが第一の仕事である。

ところで，この種の解法には従来から「最小自乗法近似」が多く用いられているが，ここでは与えられた問題を「$\sum$ (実測値－計算値)^2→最小にすること」と定義する．さらに作成すべき数学モデルは非線形であると考える．その結果，計算過程は第３図で示したごとく反復計算になる（図で $\vec{A}\ (A_1, A_2, \cdots\cdots, A_N)$ を求めることと数学モデルを作成することとは同一である)。

他方，具体的な解法として

1) 最小自乗法

2) こう配法



第３図　反復過程の一般的フローチャート

Fig. 3. General flow chart of iteration technique

3) 直 接 法

が，種々検討した結果，使用可能であることがわかった．ここでは，上記問題のために開発したこれらの解法のアルゴリズムを示す。

1. 使用する式の記述，およびその目的

$$\left.\begin{aligned} y_i &= f(A_1, A_2, \cdots\cdots, A_N, \vec{X}_i) \\ &\equiv f_i(A_1, A_2, \cdots\cdots, A_N) \\ \vec{X}_i &\equiv (X_i^{(1)}, X_i^{(2)}, \cdots\cdots, X_i^{(k)}) \end{aligned}\right\} \cdots\cdots\cdots\cdots(1)$$

ここで y_i は観測値 $\vec{X}_i$ を用いて，作成した数学モデルから求めた計算値，$A_1, A_2, \cdots\cdots, A_N$ は各解法を用いて求めるべきパラメータとする．なお $\overline{y}_i$ は $\vec{X}_i$ における観測値とする。

目 的

「各解法を用いて式(2)で定義される関数 $R(A_1, A_2, \cdots\cdots, A_N)$ を最小にするような各 $A_1, A_2, \cdots\cdots, A_N$ を求めること」。

$$R(A_1, A_2, \cdots\cdots, A_N) = \sum_{i=1}^{k} (\overline{y}_i - y_i)^2 \cdots\cdots\cdots\cdots(2)$$

2. 最 小 自 乗 法

式(2)から

$$R(A_1, A_2, \cdots\cdots, A_N) = \sum_{i=1}^{k} [\overline{y} - f_i(A_1, A_2, \cdots\cdots, A_N)]^2 \cdots\cdots\cdots\cdots(3)$$

$(A_1^{(0)}, \cdots\cdots A_N^{(0)})$ を $(A_1, A_2, \cdots\cdots, A_N)$ の第0次近似値とし，$(A_1^{(l)}, \cdots\cdots A_N^{(l)})$ がわれわれの目的を満足する値とする。

逐次近似法により求める第 n 次近似値 $\vec{A}^{(n)}$ は n を大きくしていった時，$\vec{A}^{(l)}$ に収束するものとする．(この収束性は第0次近似値 $\vec{A}^{(0)}$ によって左右されるが，一応 $\vec{A}^{(n)}$ が $\vec{A}^{(l)}$ になるように $(n \to \infty$ の時)，われわれは経験または考察により $\vec{A}^{(0)}$ を決定できるものとする)．すなわち，

$$\vec{A}^{(n+1)} = \vec{A}^{(n)} + \Delta\vec{A}^{(n)}, \quad (n=0, 1, 2, \cdots\cdots) \cdots(4)$$

から求められる $\vec{A}^{(n+1)}$ は

$$\lim_{n\to\infty} |\vec{A}^{(n+1)} - \vec{A}^{(n)}| \fallingdotseq 0, \quad \lim_{n\to\infty} \vec{A}^{(n)} \fallingdotseq A^{(l)}$$ になるものとする。

最小自乗法を用いることとは，式(4)を用いて逐次に $\{A_i^{(n)}\}$ を求めるために $\Delta A_i^{(n)}$ に関して正規方程式を作成し，それより $\Delta A_i^{(n)}$ を求め，次数の一つ高い近似値を求める手順を行なうことと同一である。

すなわち

$$R(A_1^{(n+1)}, \cdots\cdots, A_N^{(n+1)}) = \sum_{i=1}^{k} [\overline{y}_i - f_i(A_1^{(n)} + \Delta A_1^{(n)}, A_2^{(n)} + \Delta A_2^{(n)}, \cdots\cdots, A_N^{(n)} + \Delta A_N^{(n)})]^2 \cong \sum_{i=1}^{k} \left[\overline{y}_i - f_i(A_1^{(n)}, A_2^{(n)}, \cdots\cdots, A_N^{(n)}) - \left\{\left(\frac{\partial f_i}{\partial A_1}\right)_{(n)} \Delta A_1^{(n)} + \left(\frac{\partial f_i}{\partial A_2}\right)_{(n)} \Delta A_2^{(n)} + \cdots\cdots + \left(\frac{\partial f_i}{\partial A_N}\right)_{(n)} \Delta A_N^{(n)}\right\}\right]^2 \cdots\cdots(5)$$

ここで

$$\left(\frac{\partial f_i}{\partial A_j}\right)_{(n)} = \left(\frac{\partial f_i}{\partial A_j}\right)_{A_1=A_1^{(n)}, A_2=A_2^{(n)}, \cdots\cdots, A_N=A_N^{(n)}}$$

いま $B_i \equiv \overline{y}_i - f_i(A_1^{(n)}, \cdots\cdots, A_N^{(n)})$

$$C_{ij} = \left(\frac{\partial f_i}{\partial A_j}\right)_{(n)} \quad (j=1, 2, \cdots\cdots, N)$$

とすると，式(5)は式(6)のように変形できる。

$$R(\vec{A}^{(n+1)}) \fallingdotseq \sum_{i=1}^{k} (B_i - C_{i1}\Delta A_1^{(n)} - C_{i2}\Delta A_2^{(n)} - \cdots\cdots - C_{iN}\Delta A_N^{(n)})^2 \cdots\cdots\cdots\cdots(6)$$

式(6)が最小値を持つためには，

$$\frac{\partial R}{\partial(\Delta A_j^{(n)})} = 0 \quad (j=1, 2, \cdots\cdots, N) \cdots\cdots(7)$$

式(7)から次の正規方程式が作成できる。

$$\frac{\partial R}{\partial(\Delta A_1^{(n)})} = 2\sum_{i=1}^{k} C_{i1} (B_i - C_{i1}\Delta A_1^{(n)} - C_{i2}\Delta A_2^{(n)} - , \cdots\cdots - C_{iN}\Delta A_N^{(n)}) = 0$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\frac{\partial R}{\partial(\Delta A_N^{(n)})} = 2\sum_{i=1}^{k} C_{iN} (B_i - C_{i1}\Delta A_1^{(n)} - C_{i2}\Delta A_2^{(n)} - , \cdots\cdots - C_{iN}\Delta A_N^{(n)}) = 0$$

ゆえに

$$\begin{vmatrix} \sum (C_{i1})^2, & \sum (C_{i1}\cdot C_{i2}) \cdots\cdots & \sum (C_{i1}\cdot C_{iN}) \\ \sum (C_{i1}\cdot C_{i2}), & \vdots & \vdots \\ \sum (C_{i1}\cdot C_{iN}), & \vdots & \sum (C_{iN})^2 \end{vmatrix} \times \begin{vmatrix} \Delta A_1^{(n)} \\ \vdots \\ \Delta A_N^{(n)} \end{vmatrix} = \begin{vmatrix} \sum B_i C_{i1} \\ \vdots \\ \sum B_i C_{iN} \end{vmatrix} \cdots\cdots\cdots\cdots(8)$$

式(8)は正規方程式であるから，これから $(\Delta A_1^{(n)}, \cdots\cdots \Delta A_N^{(n)})$ を求める．その結果，(4)式から $\vec{A}^{(n+1)}$ が求められる。

この手順をくりかえすことにより $\{A_i^{(n)}\}$ $(i=1, \cdots\cdots N)$，$(n=0, 1, 2, \cdots\cdots)$ が求められ，この列は $\{A_i^{(l)}\}$ に収束する。

3. こ う 配 法

最小自乗法では，極値においてその微係数が0になるという条件から正規方程式を作成したが，この解法では上述の条件は使用するが，正規方程式は作成せずに計算をすすめる。

なお，この場合は問題を次のように変形する。

$(\Delta A_1^{(n)})^2 + (\Delta A_2^{(n)})^2 + \cdots\cdots + (\Delta A_N^{(n)})^2 = K_1^2$ のもとに，$R(A_1^{(n)}, A_2^{(n)}, \cdots\cdots, A_N^{(n)})$ を最小にすること．ただし $A_i^{(n)} = A_i^{(n+1)} + \Delta A_i^{(n)}$ である。

このような束縛条件を持った極値問題は，ラグランジ未定係数法を用いて，条件のない極値問題にすることができる．それが式(9)である。

$$O(\vec{A}_1^{(n)})=R(\vec{A}^{(n)})+\lambda[(\Delta A_1^{(n)})^2+(\Delta A_2^{(n)})^2+\cdots\cdots+(\Delta A_N^{(n)})^2]\rightarrow\min \cdots(9)$$

ここでλはラグランジュの未定係数である。

式(9)に $\vec{A}^{(n)}=\vec{A}^{(n+1)}+\Delta\vec{A}^{(n)}$ を代入し $\vec{A}^{(n+1)}$ のまわりでテイラ展開し，$R(\vec{A}^{(n)})$ に関しては $\Delta A_i^{(n)}$ の一次以上の項を無視する。

すなわち

$$\begin{aligned}O(A_1^{(n)},\cdots\cdots A_N^{(n)})&=R(A_1^{(n+1)}+\Delta A_1^{(n)},\cdots\cdots,A_N^{(n+1)}+\Delta A_N^{(n)}\\&\quad+\lambda([\Delta A_1^{(n)}]^2+\cdots\cdots+[\Delta A_N^{(n)}]^2)\\&\cong R(A_1^{(n+1)},\cdots\cdots,A_N^{(n+1)})+\left(\frac{\partial R}{\partial A_1}\right)_{(n+1)}\Delta A_1^{(n)}+\cdots\cdots+\left(\frac{\partial R}{\partial A_N}\right)_{(n+1)}\Delta A_N^{(n)}\\&\quad+\lambda[\Delta A_1^{(n)}]^2+\cdots\cdots+[\Delta A_N^{(n)}]^2)\end{aligned}$$

ところで第 $(n+1)$ 次近似値はまだ未定であるので，これを第 (n) 次近似値でおきかえる。

$$O(A_1^{(n)},\cdots\cdots,A_N^{(n)})\cong R(A_1^{(n)},\cdots\cdots,A_N^{(n)})+\sum_{i=1}^{N}\left(\frac{\partial R}{\partial A_i}\right)_{(n)}\Delta A_i^{(n)}+\lambda\sum_{i=1}^{N}(\Delta A_i^{(n)})^2 \cdots(10)$$

式(10)が極値を持つためには

$$\frac{\partial O}{\partial(\Delta A_i^{(n)})}=0 \quad (i=1,2,\cdots\cdots,N) \cdots\cdots(11)$$

が必要である。

式(11)から

$$\left(\frac{\partial R}{\partial A_i}\right)_{(n)}+2\lambda(\Delta A_i^{(n)})=0 \quad (i=1,2,\cdots\cdots,N) \cdots\cdots(12)$$

ゆえに

$$(\Delta A_i^{(n)})=-\frac{1}{2\lambda}\left(\frac{\partial R}{\partial A_i}\right)_{(n)} \cdots\cdots(13)$$

$\sum_{i=1}^{N}(\Delta A_i^{(n)})^2=K_1^2$ から

$$\therefore\frac{1}{4\lambda^2}\sum_{i=1}^{N}\left(\frac{\partial R}{\partial A_i}\right)_{(n)}^2=K_1^2 \cdots\cdots(14)$$

最小値を求める問題であることを考慮して，

$$\frac{1}{2\lambda}=-K_1\left[\sum_{i=1}^{N}\left(\frac{\partial R}{\partial A_i}\right)_{(n)}^2\right]^{-1/2} \cdots\cdots(15)$$

式(13)と式(15)から

$$(\Delta A_i^{(n)})=K_1\left(\sum_{i=1}^{N}\left[\frac{\partial R}{\partial A_i}\right]_{(n)}^2\right)^{-1/2}\times\left(\frac{\partial R}{\partial A_i}\right)_{(n)}(i=1,\cdots\cdots,N) \cdots\cdots(16)$$

これから次の逐次近似公式がえられる。

$$\left.\begin{aligned}A_i^{(n+1)}&=A_i^{(n)}-\Delta A_i^{(n)}\\\Delta A_i^{(n)}&=\alpha_n\cdot\left(\frac{\partial R}{\partial A_i}\right)_{(n)}\end{aligned}\right\}\cdots\cdots(17)$$

ここで

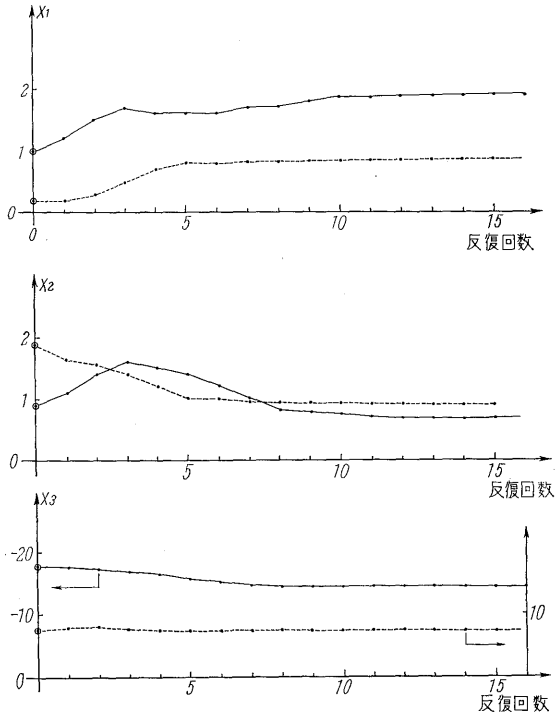
$$\left.\begin{aligned}\alpha_n&=K_1\left(\sum_{i=1}^{N}\left[\frac{\partial R}{\partial A_i}\right]_{(n)}\right)^{-1/2}\\K_1^2&=\sum_{i=1}^{N}(\Delta A_i^{(n)})^2\\R&=\sum_{i=1}^{K}(\bar{y}_i-f_i[A_1,A_2,\cdots\cdots,A_N])^2\end{aligned}\right\}\cdots(18)$$

なお，この方法は，Box らによるモデル修正法と同一なものになっていることに注目されたい。

4. 直　接　法

目的が関数 $R(A_1, A_2,\cdots\cdots, A_N)$ の最小値を求めることであり，関数 R が $(A_1, A_2,\cdots\cdots, A_N)$ に対して連続ならば任意の区間内で必ず最大値および最小値があることがわかっているから，最も原始的な方法は，各 $(A_1, A_2,\cdots\cdots, A_N)_i$ に対して関数値 R を計算し，その値の小さくなる方向に $(A_1\cdots\cdots A_N)_i$ を変えていけば必ず最小値がえられるはずである．この列挙法を能率よく行なうことにより確実に最小値を求めることができる。

すなわち，$R(A_1, A_2,\cdots\cdots, A_N)$ が $(A_1,\cdots\cdots, A_N)$ の連続関数であるならば各 $(\vec{A})_j$，$(j=1,\cdots\cdots)$ に対する関数値 $R_1, R_2,\cdots\cdots$ が，$R_1>R_2>\cdots\cdots>R_N>,\cdots\cdots$ なるように点 $(\vec{A})_j$ を求めてゆくことにより必ず最小値が求められる。



第4図　直接法による数学モデル $F(X_1, X_2, X_3)$ の作成過程

Fig. 4. Curve fitting $[F(X_1, X_2, X_3)]$ by direct search

ただし極小値が2個以上存在する場合には，初期値をいくとおりか変えて同じ計算をさせなければならない。

第4図は直接法を用いて式(19)なる数学モデルを作成した時に，求めるべきパラメータ X_1, X_2, X_3 が仮定値（初期値）から出発してどのように動きまわって収束していくかを示したものである。

$$y_i = X_3 \frac{\left(1.0 + X_1 \times e^{-\frac{Ti}{X_2}} - X_2 \times e^{-\frac{Ti}{X_2}}\right)}{(X_2 - X_1)} \quad \cdots(19)$$

与えられるデータは (T_i, y_i), $(i=1, 2, \cdots\cdots, N)$ で求めるパラメータは X_1, X_2, X_3, である．このデータは二次おくれを示す装置から得られたもので，式(19)を作成することにより，この系の時定数を求めることを目的としている。

同図では，各二とおりの異なったデータから2種類のモデルを作成した時の計算過程である．反復回数としてあるのは，任意の出発点から，何回ジクザクをしながら目的値に達したかを示すもので，大略 10 回ぐらいの行程でほぼ最小値の近くに到達していることがわかる。

IV. 直接法に対する考察と修正直接法

ところで日常経験する最適化計算では，次の性質をもった式群での極値を求める場合が多い。

1) 関数が非線形であること。
2) 関数が多変数であること。
3) 関数の独立変数がさらに関数になっており，その関数がベッセル関数のような特殊なものからなりたっていること。
4) 独立変数の変域の他に，これら変数間に条件式があること。
5) 極値が停留点にあるかどうかは全く未知であること．さらに関数に関する情報が充分でないこと。

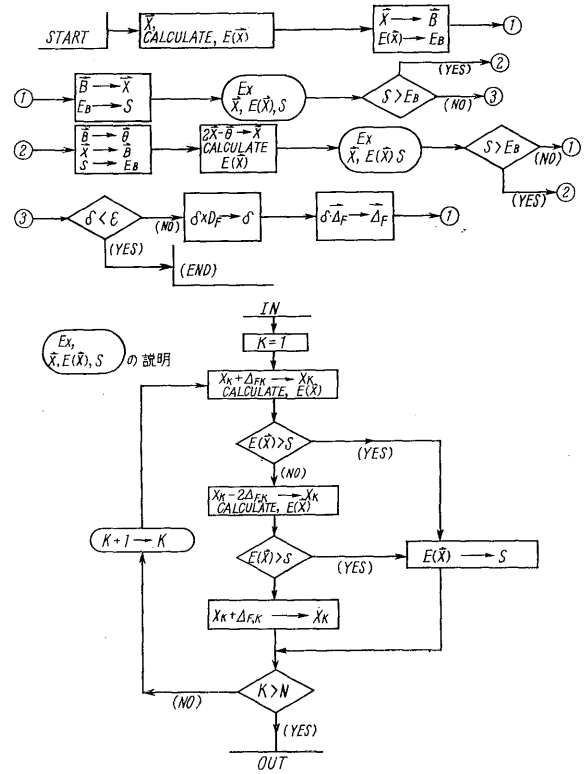
この種の問題を電子計算機を用いて解くには，できるだけ容易な演算操作を持った解法が望ましく，その点 R. Hooke & A. Jeeves（ウェスチング社）が開発した直接法は，その数学的証明の欠けている点はともかく，計算機を用いておこなう最適化手法としては適していると考えられるところでさらに直接法を種々検討した結果

1) Exploratory move する際の計算点を少なくすること。
2) 進行間隔を可変にすること。

の2点を加味した修正直接法と考えるべき手法を開発した．ここでは，この二つの解法を記述することによってその相異点を明らかにする。

1. 直　接　法

ここでは，R. Hooke & A. Jeeves の文献から直接法の概要を説明する．また問題としては最大値問題を扱う。



第5図　直接法のフローチャート

Fig. 5. Flow chart of direct search

任意の出発点から極値点に到着する過程において，ある点から次の点に移動することを move とよび，この move として次の種類を考える（目的関数が増加した場合を成功，減少した場合を失敗と呼ぶことにする）。

1) Exploratory move

この move は着目点（一番最初は出発値），の周辺における目的関数の値を数点知ることにより，目的関数の状態を知ろうとするものである．第5図に直接法のフローチャートを示したが，この部分は，同図の下の部分に示したものである．この原理は，順次各変数を進行間隔だけ変化させ，それに伴う目的関数値の変化を知ることで，その時各変数について，一方向の変化で失敗したならば正反対な方向に変数を変化させて関数値の変化を知る．この際両方向で失敗したならばその変数は，一時着目点の値に固定しておく。

このように，N 個の独立変数よりなる目的関数では1回の Exploratory move につき最大 $2 \times N$ 個の点における関数値を計算しなければならない。

2) Pattern move

Exploratory move で得られた情報をもとにして，成功した方向に基準点を移動させるとともに，次の Exploratory move すべき着目点を求める．この基準点と着目点の相異は，まず Exploratory move により成功した

場合には，その成功方向の点が基準点となり，つぎに Exploratory move する着目点は，基準点より，さらに各変数が1ピッチだけ進んだ点になる。

この2種の move を交互に行なって順次改良された点を求めてゆくと同時に，極値に近づくにつれ進行間隔を小さくしてゆき，その進行間隔減少因子がある判定常数以下になったらストップする。

2. 修正直接法

直接法の持つ単純な操作を複雑にすることなく，次の点がくみこめないか検討した。

1) Exploratory move する際の計算点を少なくする。

2) 直接法においても過去および現在に得られた情報を利用しているがさらに利用することはないか。

そして本質的には直接法と同一に Exploratory move と Pattern move を交互に行なっているが，次の点を変更した方法を確立した．この方法は，直接法を一部変更したものであることより，修正直接法と呼ぶことにする。

1) 1回の Exploratory move において最大 $2\times N$ 個の目的関数の値を計算しなければならなかったものを $(N+1)$ 回にする。

この結果，目的関数が複雑な場合には計算時間が減少されると思われる。

第6図に修正直接法のフローチャートを示したが，Exploratory move の部分は，同図下の部分に相当する。

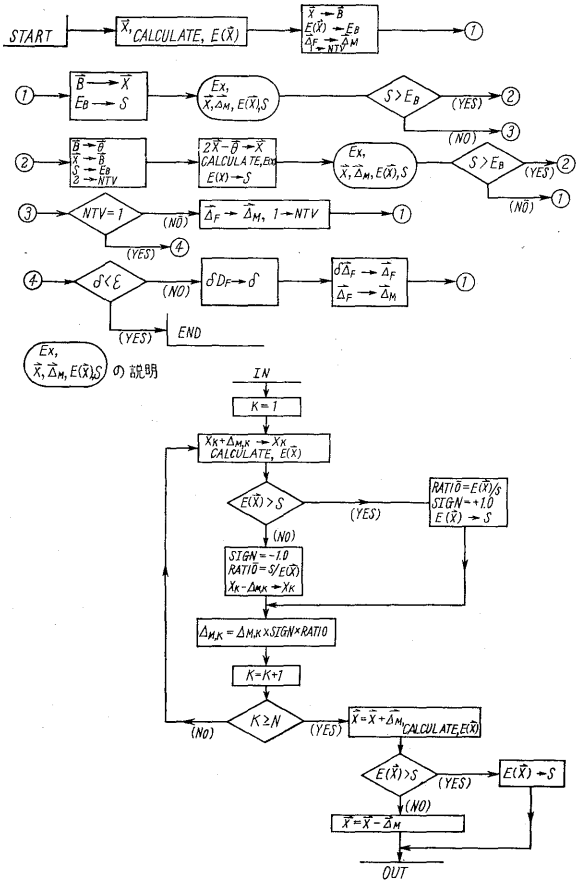
2) 直接法では Exploratory move から得られる目的関数の値を大小判別のみに使用しているが，修正直接法では大小判別に利用するとともに進行間隔を大きくする手順にもとりいれている。

すなわち，着目点と計算点における目的関数の値を比較するとともにその関数の比をとり，（常に1より大きくなるようにする）その比に +1 または −1 を乗じ，その結果に前段階で使用した進行間隔を乗じたものを次の計算に用いる進行間隔としている．このため関数値の変化の大きなところでは（一般にこの近辺では極値はありえない.）進行間隔は自動的に大きくなり，またある方向で失敗したならば，次のピッチ幅として現在のピッチ幅に −1 を乗ずるので，その方向に対しては逆に進むことになる。

このように，直接法では常に初めに定められた進行間隔で進んでいるのに対し，修正直接法では，はじめに定められた進行間隔より常に大きな幅で進み，関数値の変化に応じ，たえずその幅が変化していることになる。

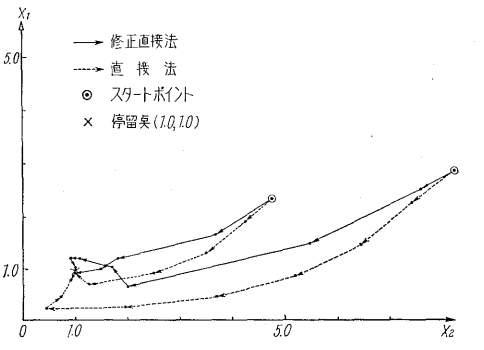
この結果，第6図に示すように，Exploratory move してすべての方向に失敗したならば極値近傍だとは考えず，いったん，はじめの定められた進行間隔に戻して，さらに Exploratory move を行なうといった手順が入っている。

なお，第5図，第6図に示した記号は次の意味を持っている。



第6図 修正直接法のフローチャート

Fig. 6. Flow chart of modified direct search



第7図 関数 $y=(0.5+0.5x_1)\cdot x_2\cdot\exp[2.0-(0.5+0.5x_1)-x_2]$ の極大値点への軌跡

Fig. 7. To find max. value of function $y=(0.5+0.5x_1)\cdot x_2\cdot\exp[2.0-(0.5+0.5x_1)-x_2]$

記　号	意　味
$\vec{X}$	$(X_1, X_2, \cdots\cdots, X_N)$, 独立変数
$E(\vec{X})$	点 $\vec{X}$ における関数値
$\vec{B}$	基準点
E_B	基準点における関数値
S	関数値を一時ストアする場所
δ	進行間隔減少因子
D_F	進行間隔減少因子を減少させるファクタ
ε	判定定数
$\vec{\Delta}_F$	$(\Delta_{F1}, \Delta_{F2}, \cdots\cdots, \Delta_{FN})$, 固定進行間隔
$\vec{\Delta}_M$	$(\Delta_{M1}, \Delta_{M2}, \cdots\cdots, \Delta_{MN})$, 可変進行間隔
N	独立変数の数
$\vec{\theta}$	前のステップの基準点
NTV, K	制御用インデックス

第7図は，$y=(0.5+0.5\,x_1)\cdot x_2 \exp[2.0-(0.5+0.5\,x_1)-x_2]$ の最大値を上記両方法で求めた場合の，出発点から最終点までの点の移動状態をしめしたものである．この関数は $x_1=x_2=1$ で最大値1を持つことがわかっており，2個の出発点をとり，その点からの移動状態を調べた．点線が直接法を用いた時の軌跡，実線が修正直接法を用いたときの軌跡である．同図から，修正直接法を用いた場合がすみやかに極値点に到着していくのがわかる。

V. FACOM 222 による線形計画法

与えられた物理系，および目的関数がすべて線形で表示される場合，その系の極値を求める手法の一つとして線形計画法がある．そこで当社にある FACOM 222 のために次に表示される系に対する同解法のコード化を行なった。

$$\left.\begin{aligned} &a_{1,1}x_1+a_{1,2}x_2,+\cdots\cdots,+a_{1,n}x_n\leq S_1\\ &a_{2,1}x_1+a_{2,2}x_2,+\cdots\cdots,+a_{2,n}x_n\leq S_2\\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ &a_{m,1}x_1+a_{m,2}x_2,+\cdots\cdots,+a_{m,n}x_n\leq S_m \end{aligned}\right\}(20)$$

$$\left.\begin{aligned} &a_{m+1,1}x_1+a_{m+1,2}x_2,+\cdots\cdots,\\ &\qquad\qquad +a_{m+1,n}x_n\geqq S_{m+1}\\ &a_{m+2,1}x_1+a_{m+2,2}x_2,+\cdots\cdots,\\ &\qquad\qquad +a_{m+2,n}x_n\geq S_{m+2}\\ &a_{m+r,1}x_1+a_{m+r,2}x_2,+\cdots\cdots,\\ &\qquad\qquad +a_{m+r,n}x_n\geq S_{m+r} \end{aligned}\right\}(21)$$

$$x_i\geq 0,\quad (i=1,2,\cdots\cdots,n)\qquad\cdots\cdots\cdots\cdots(22)$$

$$f(x_1,x_2,\cdots\cdots,x_n)=v_1x_1+v_2x_2+\cdots\cdots+v_nx_n\rightarrow max\cdots\cdots\cdots\cdots(23)$$

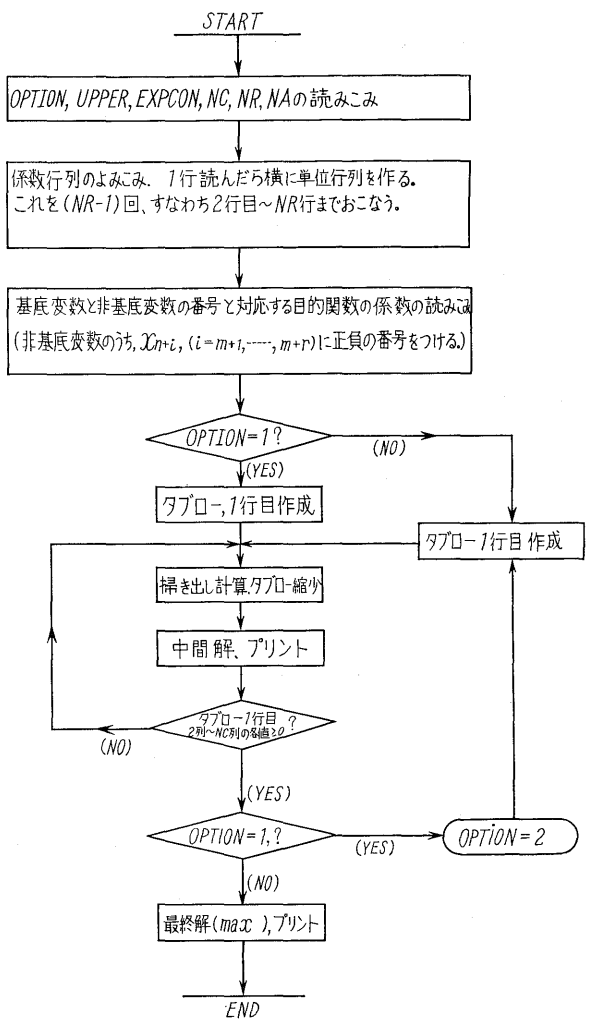
条件式が式(20)，式(22)で式(23)の極値を求める場合は，ごく標準なシンプレックス法を適用できるが，さらに式(21)が条件として加わると一般の教科書にのっている方法では，効率の良いコードは作成できない。

そこで，条件式として式(20)，式(21)，式(22)を持ち，目的関数が式(23)で与えられる場合，（もちろん，式(20)のみでも式(21)のみでも良い）に使用可能なコードを作成した。

このさい前述の理由によりシンプレックス法を前段階，後段階の二つにわけて用いた2段階解法を用いた。条件式が式(20)のみの場合は，1段階で解が求まるが，式(21)のみとか，式(20)と式(21)が混合している場合には2段階法を用いて解を求めることになる。

この2段階法とは，最終の目的関数は式(23)であるが，そのまえに，式(21)の不等式を等号になおすために導入したスラックス変数が0になるように基底解を求めるといった操作が入ることになる。

すなわち前段階の目的関数は，式(21)に用いられるスラックス変数の和であり，それが最小（具体的には0）になるような基底解を求めることである。



第8図　FACOM 222 L•P コードのフローチャート

Fig. 8.　Flow chart of L•P for FACOM 222

次にその基底解を用いて，式(23)を満足する解を求めるのが後の段階である。

第8図が，このコードの大略のフローチャートである。

同図で使用されている記号の意味は次のとおりである。

$OPTION=$ { 1.…技巧変数を含む場合，すなわち式(21)の不等式がある場合 / 2.…スラックス変数のみの場合すなわち式(20)だけの場合

$UPPER$：判定用定数，$\sim 10^{49}$

$EXPON$：最適判定$\sim 10^{-4}$

NC：タブローの列数

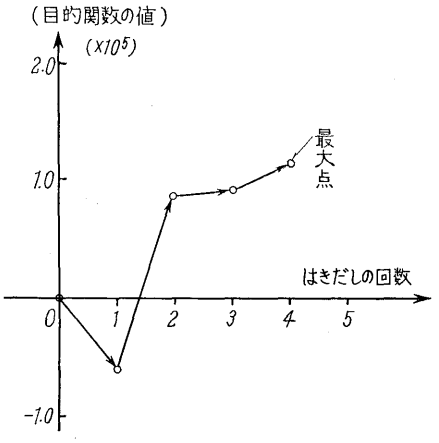
NR：タブローの行数

NA：技巧変数を含む式の数

なお，このコードは「$L\cdot P$ 25」となずけられており，主な制限条件は次のとおりである。

1） 独立変数の数≦25

2） 条件式の数 ≦30



第9図 L•P 25による計算結果

Fig. 9. Calculated result by L.P 25

本コードを用いて，条件式が7個，独立変数が10個あった問題に対する目的関数値の変化の状態を第9図に示す。

VI. あ と が き

ダイナミックプログラミング的考え方は，この種の問題に対し非常に有益な指示を示してくれるが，ここではこの考え方については何一つふれなかった。

参 考 文 献

(1) R. Bellman: "Adaptive Control Processes; a guided Tour"
(2) R. Hooke et al: "Direct Search, Solution of Numerical and statistical problems"
(3) C. Zexer: "A Mathematical Aid in Optimizing Engineering Designs"
(4) K. Klee: "Algol-compiled Reviced Simplex Algorithems for B-220"
(5) 森口ほか："線形計画法"（応用数学講座）
(6) A. Nierenberg: "A Method for Minimizing a Function of n Variables"
(7) W. Carroll: "The Created Response Surface Technique for Optimizing NonlinearRestrained Systems"
(8) 大星訳：最大傾斜法の数学的基礎
(9) P. Box & B. Wilson: "On the experimental Attainment of Optimum Conditions"
(10) R. Bellmann: "Introduction to Dynamic Programming"
(11) 日科技連セミナー：線形計画法
(12) 市川："化学工業における最適化"
(13) Charles W. Carroll: "An Approach to Optimizing Control of a Restrained System by Dynamic Gradient Technique"
(14) H. Brooks: "A Comparison of Maximum-seeking Methods"
(15) D. C. Sherman: "KAPL-2114, a Formal Procedure for Rapid Optimization of Design Performance"
(16) P. Box: "Some General Considerations in precise Optimization"
(17) 共立全書：解説自動制御
(18) 野沢：自動制御入門
(19) R. Bellmann: "Applied Dynamic Programming"