

JAERI-M

4 9 8 0

ノンリニア・プログラミングに
よる関数近似プログラム

1972年9月

鈴木 忠 和

日 本 原 子 力 研 究 所
Japan Atomic Energy Research Institute

ノンリニア・プログラミングによる関数近似プログラム

日本原子力研究所東海研究所原子炉工学部

鈴木 忠 和

(1972年8月29日受理)

先に完成したリニア・プログラミングによる関数近似プログラム〔1〕の拡張として、ノンリニア・プログラミングを用いた関数近似プログラムを完成した。ノンリニア・プログラミングによる関数近似は、リニア・プログラミングでは扱うことのできなかつた近似関数がパラメータに関して非線形な問題を扱うことができ、更に、近似の基準として任意のノルムを使用できる。

また、リニア・プログラミングの場合に付加できたパラメータや近似関数に対する種々の条件はノンリニア・プログラミングの場合においても付加できる。

Functional Approximation Program by Non-linear Programming

Tadakazu SUZUKI

Div. of Reactor Engineering, Tokai, JAERI

(Received August 29, 1972)

A functional approximation program by non-linear programming is presented, which is a generalization of the previous program by linear programming. Using this program, non-linear approximation functions belonging to arbitrary norms can be treated. The same auxiliary conditions are imposed as in the case of linear programming.

目 次

1	ノンリニア・プログラミングとGRADIENT法	1
2	関連近似への応用	5
3	プログラムの説明と入力データ	9
4	使用例	12
5	おわりに	20
6	参考文献	20

1 ノンリアプログラミングとGRADIENT 法

ノンリアプログラミングの問題は、一般には制限条件

$$\left. \begin{aligned} g_i(\mathbf{x}) &\leq b_i & i \in I \\ x_j &\geq 0 & j \in J \end{aligned} \right\} \quad (1)$$

の下で

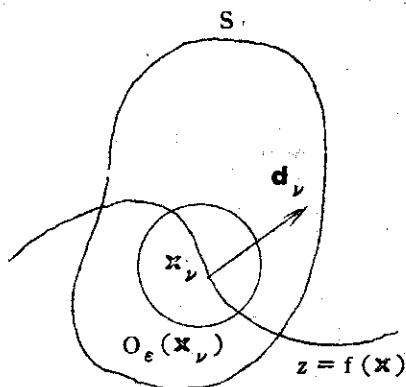
$$\max Z = f(\mathbf{x})$$

なるようにベクトル $\mathbf{x} \in E^m$ を決定すること。という形で与えられる。ここで関数 g_i , f 及び定数 b_i は既知とする。

関数 g_i , f が $\mathbf{x} = (x_1 \cdots x_m)^t$ に関して線形であれば(1)はリアプログラミングの問題となり、その場合は Simplex 法を用いて解くことができるが、非線形の場合は線形の場合における Simplex 法のような確立された解法はなく、関数 g_i , f の形により例えば問題を線形近似して Simplex 法を用いるとか、制限条件を満たす全ての解を求めて、その中から Z を最大にする点を求めるなど、問題に応じて種々の解法が開発されており、Gradient 法はその中の1つの解法である。この方法は初期実行可能率から目的関数 $f(\mathbf{x})$ を更に大きくする点を次々に求めていくくり返し法で、次の点を求める際、 $f(\mathbf{x})$ の変化率が最大な方向、即ちその点における $f(\mathbf{x})$ の gradient ベクトル方向に進んでいこうというものである。然しながら \mathbf{x} に対しては(1)の第1式や第2式のような制限があるため、常に gradient ベクトルの方向に進めるというのではなく、そのためにこの方向ではある実行可能を与えられた時に、次に進むべき方向と距離を計算することが大切な問題になる。

関数 $g_i(\mathbf{x})$, $f(\mathbf{x})$ は C^1 級とし、 $g_i(\mathbf{x})$ は凸関数とする。集合 S を $S = \{ \mathbf{x} \mid g_i(\mathbf{x}) \leq b_i, i \in I, x_j \geq 0, j \in J \}$ と定義する。ある段階で実行可能解 \mathbf{x}_ν (即ち $\mathbf{x}_\nu \in S$) が得られているとしその点における $f(\mathbf{x})$ の gradient ベクトルを $\mathbf{d}_\nu^t = \nabla f(\mathbf{x}_\nu) = (\partial f / \partial x_1 \cdots \partial f / \partial x_m)_{\mathbf{x}=\mathbf{x}_\nu}$ とする。点 \mathbf{x}_ν から進んで $f(\mathbf{x}_{\nu+1}) > f(\mathbf{x}_\nu)$, $\mathbf{x}_{\nu+1} \in S$ なる点 $\mathbf{x}_{\nu+1}$ をみつけようという訳であるが、この時次の2つの場合を考えてみよう。

(1) 点 \mathbf{x}_ν が集合 S の内点である場合(図-1) この場合 \mathbf{x}_ν の ε -近傍 $O_\varepsilon(\mathbf{x}_\nu)$ で、 $O_\varepsilon(\mathbf{x}_\nu) \subset S$ が存在する、それ故その近傍内で進むかぎり方向としては任意方向に動けるはずで



(図-1)

ある。それ故この場合は $f(\mathbf{x})$ の変化率が最大になる方向 \mathbf{d}_ν を選ぶことができ、

$$\mathbf{x}_{\nu+1} = \mathbf{x}_\nu + \lambda_\nu \mathbf{d}_\nu, \quad 0 < \lambda_\nu < \varepsilon$$

として $\mathbf{x}_{\nu+1}$ を決定できる。このとき $f(\mathbf{x}_{\nu+1}) = f(\mathbf{x}_\nu + \lambda_\nu \mathbf{d}_\nu)$ を Taylor 展開してみると、

$$f(\mathbf{x}_\nu + \lambda_\nu \mathbf{d}_\nu) = f(\mathbf{x}_\nu) + \lambda_\nu \nabla f(\mathbf{x}_\nu + \theta \lambda_\nu \mathbf{d}_\nu) \mathbf{d}_\nu \quad (0 \leq \theta \leq 1)$$

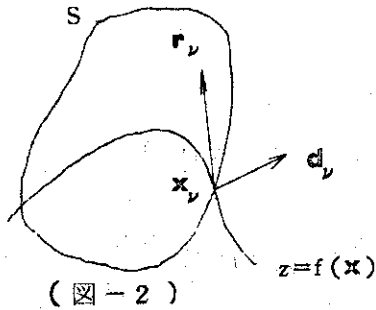
$f(\mathbf{x})$ は C^1 級としたから $\nabla f(\mathbf{x}) \mathbf{d}_\nu$ は \mathbf{x} の連続関数で $\nabla f(\mathbf{x}_\nu) \mathbf{d}_\nu > 0$ である。それ故点 \mathbf{x}_ν のある δ -近傍 $O_\delta(\mathbf{x}_\nu)$ が存在して $\mathbf{x} \in O_\delta(\mathbf{x}_\nu)$ なる全ての \mathbf{x} に対し

$\nabla f(\mathbf{x}_\nu) \mathbf{d}_\nu > 0$ である、即ち $\lambda_\nu = \min(\epsilon, \delta)$

なる λ_ν に対し $\mathbf{x}_{\nu+1} = \mathbf{x}_\nu + \lambda_\nu \mathbf{d}_\nu$ は実行可能解で $f(\mathbf{x}_{\nu+1}) > f(\mathbf{x}_\nu)$ である。

(四) \mathbf{x}_ν が S の境界面上にある場合 (図-2)

この場合、 \mathbf{d}_ν 方向に進んで得られる点が必ずしも S に含まれ得ない。即ち方向を新たに決定しなければならないことになる。



(図-2)

明らかなように \mathbf{x}_ν が S の境界面上にあるということは、ある $i \in I$ に対し

$$g_i(\mathbf{x}_\nu) = b_i$$

となっているか、ある $j \in J$ に対し $x_{\nu j} = 0$ となっており、いまそのような index の集合をそれぞれ $I(\nu), J(\nu)$ としてみよう。即ち $I(\nu) = \{i | g_i(\mathbf{x}_\nu) = b_i, i \in I\}, J(\nu) = \{j | x_{\nu j} = 0, j \in J\}$ とする。そこで方向 \mathbf{r} を決める際、今

度は \mathbf{r} は何らかの制限を受けるわけだが、それは若し gradient ベクトル方向に動いたなら $i \in I(\nu)$ に対し $g_i(\mathbf{x}) > b_i$ となったり $j \in J(\nu)$ に対し $x_j < 0$ となることからくる。いま方向を仮りに \mathbf{r}_ν とすると $\mathbf{x} = \mathbf{x}_\nu + \lambda \mathbf{r}_\nu$ が実行可能 (i.e. $\mathbf{x} \in S$) であるためには

$$\left. \begin{aligned} g_i(\mathbf{x}) = g_i(\mathbf{x}_\nu + \lambda \mathbf{r}_\nu) &\leq b_i, \quad i \in I(\nu) \\ x_j = x_{\nu j} + \lambda r_{\nu j} &\geq 0, \quad j \in J(\nu) \end{aligned} \right\} \quad (2)$$

であることが必要である。仮定により $0 < \mu < 1$ なる μ に対し

$$g_i[\mu \mathbf{x} + (1-\mu)\mathbf{x}_\nu] \leq \mu g_i(\mathbf{x}) + (1-\mu)g_i(\mathbf{x}_\nu)$$

即ち

$$\{g_i[\mu \mathbf{x} + (1-\mu)\mathbf{x}_\nu] - g_i(\mathbf{x}_\nu)\} / \mu \leq g_i(\mathbf{x}) - g_i(\mathbf{x}_\nu) \quad (3)$$

$g_i[\mu \mathbf{x} + (1-\mu)\mathbf{x}_\nu]$ を Taylor 展開すると

$$g_i[\mu \mathbf{x} + (1-\mu)\mathbf{x}_\nu] = g_i(\mathbf{x}_\nu) + \mu \nabla g_i(\mathbf{x}_\nu + \mu \theta(\mathbf{x} - \mathbf{x}_\nu)) \times (\mathbf{x} - \mathbf{x}_\nu), \quad 0 \leq \theta \leq 1$$

これを (3) に代入し整理すると

$$\nabla g_i(\mathbf{x}_\nu + \mu \theta(\mathbf{x} - \mathbf{x}_\nu)) (\mathbf{x} - \mathbf{x}_\nu) \leq g_i(\mathbf{x}) - g_i(\mathbf{x}_\nu)$$

$\mu \rightarrow 0$ とすると

$$\nabla g_i(\mathbf{x}_\nu) (\mathbf{x} - \mathbf{x}_\nu) \leq g_i(\mathbf{x}) - g_i(\mathbf{x}_\nu)$$

故に $g_i(\mathbf{x}) \leq b_i, i \in I(\nu)$ であるなら

$$\nabla g_i(\mathbf{x}_\nu) (\mathbf{x} - \mathbf{x}_\nu) = \lambda \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu \leq b_i - b_i = 0$$

また、 $x_j = x_{\nu j} + \lambda r_{\nu j} = \lambda r_{\nu j} \geq 0$ である

故に (2) は

$$\left. \begin{aligned} \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu &\leq 0 & i \in I(\nu) \\ r_{\nu j} &\geq 0 & j \in J(\nu) \end{aligned} \right\} \quad (4)$$

と同値である。

それ故 $f(\mathbf{x})$ の変化率が最大になるような方向 \mathbf{r} は

$$\left. \begin{aligned} \nabla g_i(\mathbf{x}_\nu) \mathbf{r} &\leq 0, & i \in I(\nu) \\ r_j &\geq 0, & j \in J(\nu) \end{aligned} \right\} \quad (5)$$

の下で

$$\max D_{\mathbf{r}} f(\mathbf{x}_\nu)$$

という問題を解いて得られる

$D_{\mathbf{r}} f(\mathbf{x}_\nu)$ は $f(\mathbf{x})$ の点 \mathbf{x}_ν における \mathbf{r} 方向の変化率で、 $|\mathbf{r}| = 1$ とすると ($|\mathbf{r}|$ は \mathbf{r} の内積)

$$D_{\mathbf{r}} f(\mathbf{x}_\nu) = \lim_{h \rightarrow 0} \{ [f(\mathbf{x}_\nu + h\mathbf{r}) - f(\mathbf{x}_\nu)] / h \} \quad (6)$$

$f(\mathbf{x}_\nu + h\mathbf{r})$ を Taylor 展開すると

$$f(\mathbf{x}_\nu + h\mathbf{r}) = f(\mathbf{x}_\nu) + \nabla f[\theta \mathbf{x}_\nu + (1-\theta)(\mathbf{x}_\nu + h\mathbf{r})] \cdot h\mathbf{r}$$

但し、 $0 \leq \theta \leq 1$ 。これを(6)に代入すると

$$\begin{aligned} D_{\mathbf{r}} f(\mathbf{x}_\nu) &= \lim_{h \rightarrow 0} \nabla f[\theta \mathbf{x}_\nu + (1-\theta)(\mathbf{x}_\nu + h\mathbf{r})] \cdot \mathbf{r} \\ &= \nabla f(\mathbf{x}_\nu) \mathbf{r} \end{aligned}$$

それ故(5)は

$$\left. \begin{aligned} \nabla g_i(\mathbf{x}_\nu) \mathbf{r} &\leq 0 & i \in I(\nu) \\ r_j &\geq 0 & j \in J(\nu) \\ |\mathbf{r}| &= 1 \end{aligned} \right\} \quad (7)$$

の下で $\max \nabla f(\mathbf{x}_\nu) \mathbf{r}$

とすることができる。

(7)の問題は、条件 $|\mathbf{r}| = 1$ を省けば \mathbf{r} に関して線形で、 $|\mathbf{r}| = 1$ を線形近似してやると Simplex 法を用いて解けそうである。Zoutendijk [2]はこの条件を次のように近似している。

$$\left. \begin{aligned} 0 \leq r_j \leq 1 & & j \in J(\nu) \\ -1 \leq r_j \leq 1 & & j \in \bar{J}(\nu) \end{aligned} \right\} \quad (8)$$

この近似により我々は Simplex 法を用いて \mathbf{r} を決定することができる。

以上で、実行可能解 \mathbf{x}_ν が与えられた時、次の実行可能解 $\mathbf{x}_{\nu+1}$ を決める方向を決定するフローが訳ったが、(1)の場合、即ち \mathbf{x}_ν が S の内点になっている場合、(4)における $I(\nu)$ と $J(\nu)$ が空集合になることを考慮すれば、(4)における方向を決定する(7)式は(1)においても適用できることが訳る。即ちその場合 $I(\nu) = J(\nu) = \emptyset$ だから、(7)に対応する式として

条件 $|\mathbf{r}| = 1$

の下で $\max \nabla f(\mathbf{x}_\nu) \mathbf{r}$

$$\begin{aligned} \nabla f(\mathbf{x}_\nu) \mathbf{r} &= |\nabla f(\mathbf{x}_\nu)| \cdot |\mathbf{r}| \cos \theta \\ &= |\nabla f(\mathbf{x}_\nu)| \cos \theta \quad (\because |\mathbf{r}| = 1) \end{aligned}$$

(但し θ はベクトル $\nabla f(\mathbf{x}_\nu)$ と \mathbf{r} の成す角)

故に $\nabla f(\mathbf{x}_\nu) \mathbf{r}$ の最大値は $\cos \theta = 1$ のとき、即ち \mathbf{r} が $\nabla f(\mathbf{x}_\nu)$ と一致する時であることがわかる。それ故我々は実際の計算においては(1)の場合を考える必要がない訳である。

さて、これで Gradient 法のポイントである方向を決定できるが、次に距離 λ を決定しなければならぬ。この λ を制限するものは、点 $\mathbf{x}_{\nu+1}$ が実行可能でなければならぬということである。即ち

$$\left. \begin{array}{l} g_i(\mathbf{x}_{\nu+1}) \leq b_i \quad i \in I \\ x_{\nu+1j} \geq 0 \quad j \in J \end{array} \right\} \quad (9)$$

$g_i(\mathbf{x}_{\nu+1}) = g_i(\mathbf{x}_\nu + \lambda \mathbf{r}_\nu)$ を Taylor 展開すると

$$g_i(\mathbf{x}_\nu + \lambda \mathbf{r}_\nu) = g_i(\mathbf{x}_\nu) + \lambda \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu + \theta \lambda^2 \nabla^2 g_i(\mathbf{x}_\nu) \mathbf{r}_\nu \quad (0 \leq \theta \leq 1)$$

$$\theta \rightarrow 0 \text{ とすると (9) の第 1 式は } g_i(\mathbf{x}_{\nu+1}) = g_i(\mathbf{x}_\nu) + \lambda \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu \leq b_i$$

$$\therefore \lambda \leq (b_i - g_i(\mathbf{x}_\nu)) / \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu, \quad i \in I \quad (10)$$

\mathbf{x}_ν は実行可能解であったから、 $b_i \geq g_i(\mathbf{x}_\nu)$ である、それ故 $\lambda > 0$ であるためには $\nabla g_i(\mathbf{x}_\nu) \mathbf{r} > 0$ でなければならぬ。また(9)の第2式より

$$\begin{aligned} x_{\nu+1j} = x_{\nu j} + \lambda r_{\nu j} &\geq 0 \\ \therefore \lambda &\leq -x_{\nu j} / r_{\nu j}, \quad j \in J \end{aligned} \quad (11)$$

(10), (11) より λ のとり得る範囲は

$$\begin{aligned} \gamma = \min_i \{ (b_i - g_i(\mathbf{x}_\nu)) / \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu, \\ \nabla g_i(\mathbf{x}_\nu) \mathbf{r}_\nu \geq 0, \quad i \in I \} \end{aligned} \quad (12)$$

$$\rho = \min_j \{ -x_{\nu j} / r_{\nu j}, \quad j \in J \} \quad (13)$$

とすると $0 < \lambda < \min(\gamma, \rho)$ である。そしてこの範囲内の値で目的関数 $Z = f(\mathbf{x}_\nu + \lambda \mathbf{r}_\nu)$ が最大になるような値を λ としてとればよいことになる。

以上で我々は実行可能解 \mathbf{x}_ν が得られた時、 $f(\mathbf{x}_{\nu+1}) > f(\mathbf{x}_\nu)$ 、 $\mathbf{x}_{\nu+1} \in S$ なる $\mathbf{x}_{\nu+1} = \mathbf{x}_\nu + \lambda \mathbf{r}_\nu$ を決めることができるわけで、このくり返しにより実行可能解の列、

$\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_\nu, \dots\}$ (無限列かもしれぬ) を求めていけば良い。若しある段階で(7)の $\nabla f(\mathbf{x}_\nu) \mathbf{r}_\nu$ が負になったなら、どの方向に進んでも更に $f(\mathbf{x})$ を大きくし得る点はないわけで、その時得られた点 \mathbf{x}_ν が求める点である。

2 関数近似への応用

ノンリニアプログラミングの関数近似問題への応用は考え方としてはリニアプログラミングの場合〔3〕と全く同じで、ただ $g_j(\mathbf{x})$ や $f(\mathbf{x})$ が \mathbf{x} に関して線形である必要がないため、誤差ノルムとして一般に l_p ノルムを考慮することができる。

さて問題の定式化を考えてみよう。

$(t_j, y_j), j=1 \dots n$ を与えられたデータ対とし、これをある連続関数 $g(\mathbf{x}; t)$ で l_p ノルムの意味で残差が最小になるように近似する、即ち

$$\min \|g(\mathbf{x}; t_j) - y_j\|_p$$

なるようにパラメータ $\mathbf{x} \in E^m$ を決定しようという訳である。我々は誤差ノルムとして l_1, l_2, l_∞ の3つを考慮することにする。即ち

$$\begin{aligned} \min \|g(\mathbf{x}; t_j) - y_j\|_1 &= \min \sum_j |g(\mathbf{x}; t_j) - y_j| \\ \min \|g(\mathbf{x}; t_j) - y_j\|_2 &= \min \sum_j \{g(\mathbf{x}; t_j) - y_j\}^2 \\ \min \|g(\mathbf{x}; t_j) - y_j\|_\infty &= \min \{ \max_j |g(\mathbf{x}; t_j) - y_j| \} \end{aligned}$$

なるように \mathbf{x} を決定する。

P. LAFATA と J. B. ROSEN〔4〕により与えられた方法を用いて我々は容易にこれをノンリニアプログラミングの問題にすることができる。即ち

$$\left. \begin{aligned} Cg(\mathbf{x}; t_j) - \gamma_j &\leq Cy_j \quad j=1 \dots N (=2n) \\ \text{の下で } \max\left(-\sum_j^N \gamma_j\right) & \text{ (} l_1 \text{) or } \max\left(-\sum_j^N \gamma_j^2\right) \text{ (} l_2 \text{)} \end{aligned} \right\} \quad (14)$$

但し $t_j = t_{n+j}, \gamma_j = \gamma_{n+j} (j=1 \dots n), C=1 (j=1 \dots n), -1 (j=n+1 \dots N)$

または

$$\left. \begin{aligned} Cg(\mathbf{x}; t_j) - \gamma &\leq Cy_j \quad j=1 \dots N (=2n) \\ \text{の下で } \max(-\gamma) & \text{ (} l_\infty \text{)} \end{aligned} \right\} \quad (15)$$

なるように $\mathbf{X} = (\mathbf{x}, \gamma) \in E^{m+1}$ または $\mathbf{X} = (\mathbf{x}, \gamma_1, \dots, \gamma_n) \in E^{m+n}$ を決定する。

問題を一般化するために

$$G_j(\mathbf{X}) = \begin{cases} Cg(\mathbf{x}; t_j) - \gamma_j & (l_1 \text{ or } l_2) \\ Cg(\mathbf{x}; t_j) - \gamma & (l_\infty) \end{cases}, j=1 \dots N$$

$$b_j = Cy_j, \quad j=1 \dots N$$

$$f(\mathbf{X}) = \begin{cases} -\sum_j^N \gamma_j & (l_1) \\ -\sum_j^N \gamma_j^2 & (l_2) \\ -\gamma & (l_\infty) \end{cases}$$

としよう。すると問題は、

条件

$$\left. \begin{aligned} G_i(\mathbf{X}) &\leq b_i & i=1 \dots N \\ x_j &\geq 0 & j \in J \end{aligned} \right\} \quad (16)$$

の下で

$$\max Z = f(\mathbf{X})$$

とすることができる。但し上の $x_j \geq 0, j \in J$ なる条件は、近似関数 $g(\mathbf{x}; t)$ のパラメータ $\mathbf{x} = (x_1 \cdots x_m)$ の中で正でなければならないという条件が課せられる場合に、その index 集合を J とする。例えば、 $g(\mathbf{x}; t)$ の中に $\sqrt{ax_j}$ という項があったなら、($a > 0$ とする) 明らかに $x_j \geq 0$ なる条件が必要で、 J の中に index j が入っていないといけない。計算を始めるのに、初期実行可能解 $\mathbf{X}_0 = (\mathbf{x}_0, \gamma_0)$ が必要であるが、若し \mathbf{x}_0 の初期推定値が既知であれば、それを入力し、未知であれば $\mathbf{x}_0 = 0$ として、 γ_0 または $(\gamma_{01} \cdots \gamma_{0n})$ を次の式により計算する。

$$\gamma_{0j} = \max_j \{ |g(\mathbf{x}_0; t_j) - y_j| \} + 1 \quad (l_\infty)$$

$$\gamma_{0j} = |g(\mathbf{x}_0; t_j) - y_j| + 1, j = 1 \cdots n \quad (l_1, l_2)$$

これにより初期実行可能解 $\mathbf{X}_0 \in S$ が得られる。さて方向 \mathbf{r} の決定は前節の(7)より計算できるが、その前に $I(\nu), J(\nu)$ を調べておく必要がある。

即ち、 $I(\nu) = \{ i | G_i(\mathbf{X}_0) - b_i \geq 0, i = 1 \cdots N \}$

$J(\nu) = \{ j | X_{0j} \leq 0, j \in J \}$

とするのである。そして(7), (8)より

$$\left. \begin{aligned} \nabla G_j(\mathbf{X}_0) \mathbf{r} &\leq 0, & i \in I(\nu) \\ 0 \leq r_j &\leq 1, & j \in J(\nu) \\ -1 \leq r_j &\leq 1, & j \in \bar{J}(\nu), j \in J \end{aligned} \right\} \quad (9)$$

$$\max Z = \nabla f(\mathbf{X}_0) \mathbf{r}$$

から Simplex 法により \mathbf{r} を決定する。但し、 $I(\nu)$ が空集合の時はすぐに $\mathbf{r} = \nabla f(\mathbf{X}_0)$ とする。我々の問題では

$$\nabla f(\mathbf{X}_0) = \begin{cases} (0 \cdots 0 \ -1) & (l_\infty) \\ (0 \cdots 0 \ -1 \cdots -1) & (l_1) \\ (0 \cdots 0 \ -2\gamma_{01} \cdots -2\gamma_{0n}) & (l_2) \end{cases}$$

である。 \mathbf{r} が得られた段階で終結条件を吟味しなければならない。即ち $\nabla f(\mathbf{X}_0) \mathbf{r}$ の正負を調べ、 > 0 ならば更に目的関数を大きくする点が存在するので計算を続行し、 ≤ 0 であれば、既にその時求まっている \mathbf{X}_0 が最適解である。

$\nabla f(\mathbf{X}_0) \mathbf{r} > 0$ の時、次に(9), (10)より λ の上限を計算する。いま $\alpha = \min(\gamma, \rho)$ が計算されたものとする。このとき $0 < \lambda \leq \alpha$ で $f(\mathbf{X}_1) = f(\mathbf{X}_0 + \lambda \mathbf{r})$ を最大にするように λ を決定しなければならない。これは $f(\mathbf{X}_0 + \lambda \mathbf{r})$ を λ の関数とみなすと、 $0 < \lambda \leq \alpha$ の範囲で Z を最大にする点を求める問題となる。

我々の場合 $f(\mathbf{X}) = \begin{cases} -\sum_j \gamma_j & (l_1) \\ -\sum_j \gamma_j^2 & (l_2) \\ -\gamma & (l_\infty) \end{cases}$

であったから、

$$f(\mathbf{X}_0 + \lambda \mathbf{r}) = \begin{cases} (-\sum_j r_{m+j})\lambda + (-\sum_j r_{oj}) & (l_1) \\ (-\sum_j r_{m+j}^2)\lambda^2 + 2(-\sum_j r_{m+j} r_{oj})\lambda + (-\sum_j r_{oj}^2) & (l_2) \\ (-r_{m+1})\lambda + (-r_o) & (l_\infty) \end{cases}$$

即ち、 l_1, l_∞ ノルムの場合は λ に関する一次関数、 l_2 ノルムの場合は二次関数となる。そして l_1, l_∞ の場合は、 $\nabla f(\mathbf{X}_0)\mathbf{r} > 0$ なる条件より $-\sum_j r_{m+j} > 0$ または $-r_{m+1} > 0$ だから f は λ に関して単調に増加する。それ故 $0 < \lambda \leq \alpha$ で $f(\mathbf{X}_0 + \lambda \mathbf{r})$ を最大にする点は $\lambda = \alpha$ であることが訳る。

次に l_2 の時、 λ の二乗の係数は常に < 0 だからグラフは上に凸な曲線となる。

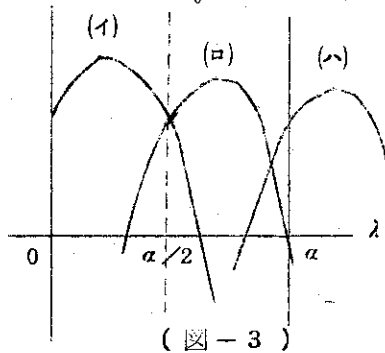
いま $C_1 = -\sum_j r_{m+j}^2$

$C_2 = -\sum_j r_{m+j} r_{oj}$

$C_3 = -\sum_j r_{oj}^2$

とすると

$$f(\mathbf{X}_0 + \lambda \mathbf{r}) = C_1 \lambda^2 + 2C_2 \lambda + C_3$$



$$= C_1 (\lambda + C_2/C_1)^2 + C_3 - C_2^2/C_1$$

条件 $\nabla f(\mathbf{X}_0)\mathbf{r} = -\sum_j r_{m+j} r_{oj} > 0$ より $C_2 > 0$ である。それ故、グラフの対称軸 $\lambda = -C_2/C_1$ は常に λ の正の側にあることがわかる。そしてこの対称軸の位置によって

$f(\mathbf{X}_0 + \lambda \mathbf{r})$ の最大値は異なる。即ち

$$0 < -C_2/C_1 < \alpha \quad (\text{図-3(イ), (ロ)})$$

なら Z は点 $-C_2/C_1$ で最大となり、 $\lambda = -C_2/C_1$ とする。

また、

$$\alpha \leq -C_2/C_1 \quad (\text{図-3(ハ)})$$

なら $0 < \lambda \leq \alpha$ の範囲では Z は点 α で最大となり $\lambda = \alpha$ とする。

こうして得られた λ と (7) により得られた方向 \mathbf{r} から $\mathbf{X}_1 = \mathbf{X}_0 + \lambda \mathbf{r}$ により次の点を求めて $\mathbf{X}_0 = \mathbf{X}_1$ として計算をくり返す。注意しなければならないのは、この実行可能解の列が無限列になることがあるということで、そのような場合終結条件、 $\nabla f(\mathbf{X})\mathbf{r} \leq 0$ では計算が無限ループしてしまふ。この場合、点列 $\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_\nu, \dots\}$ は必ず1つの集積点を持つことから(若し集積点を持たなければ我々の問題は非有界な解を持つことになる)

収束条件

$$|(\mathbf{X}_{\nu+1} - \mathbf{X}_\nu) / \mathbf{X}_{\nu+1}| < \epsilon$$

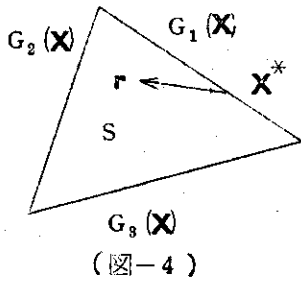
によって収束の判定をすることにする。つまり、いつも2つの方法により終結の判定をするわけである。

[注] リニアプログラミングの場合と同様にこの場合も近似関数 $g(\mathbf{x}; t)$ や、パラメータ $\mathbf{x} = (x_1 \dots x_m)^t$ に対し種々の条件を付加することが可能である。例えば単調性を要求する場合

$$a \leq \frac{\partial g(\mathbf{x}; t)}{\partial t} \Big|_{t=t_j} \leq b, \quad j = j_1 \dots j_\ell$$

と書けるからこれは(10)の第一式の中に組みこむことができるのである。

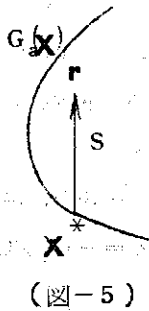
また、 $g(\mathbf{x}; t)$ と $f(\mathbf{x})$ が \mathbf{x} に関して線形の場合、リアプログミングの問題となり、Simplex法を用いて解けるわけであるが、Gradient法を用いても解くことができる。但しこの場合、距離 λ の上限を求める(12)、(13)は全ての $i \in I, j \in J$ ではなく、 $i \in I$ で $i \in I(\nu)$ 、 $j \in J$ で、 $j \in J(\nu)$ で良い。この理由は $g(\mathbf{x}; t)$ が \mathbf{x} に関し線形の場合、 S の境界面は超平面となり、その超平面上の点 \mathbf{x}^* 、即ち $i \in I(\nu)$ に対する $G_i(\mathbf{x}^*)$ からある方向で進む時、方向が(17)式で決められる限り制限式 $G_i(\mathbf{x}) \leq b_i, i \in I(\nu)$ を侵すことはないからである。直観的にみるために、いま



$S = \{ \mathbf{x} \mid G_i(\mathbf{x}) \leq b_i, i=1, 2, 3 \}$ の場合を考えてみよう。 $G_i(\mathbf{x})$ が \mathbf{x} に関して線形であれば、 S は例えば(図-4)のようになっている。ある段階での実行可能解 \mathbf{x}^* が $G_1(\mathbf{x})$ 上にあったとする。即ち

$I(\nu) = \{ 1 \}$ である。この時点 \mathbf{x}^* から方向 \mathbf{r} で次の点を求めようとする時明らかなように制限式 $G_1(\mathbf{x}) \leq b_1$ は侵される心配はない。即ち(12)は $I - I(\nu)$ について考えれば良いのである。

ところが、 $G_i(\mathbf{x})$ が \mathbf{x} に関して非線形の場合、 S の境界面は超曲面となり、例えば(図-5)のようになっている。この時 $G_2(\mathbf{x})$ 上の実行可能 \mathbf{x}^* から(17)により決定された方



向 \mathbf{r} で進むとき、またこの $G_2(\mathbf{x}) \leq b_2$ なる制限式を侵す可能性があるので、(12)においては常に $i \in I$ について考える必要がある。

線形の問題に対するSimplex法とGradient法の使用例は4節の例3と例4にあげる

3 プログラムの説明と入力データ

プログラムは主プログラムGRDNLPと7個の副プログラムから成り、それぞれ次の役割りを果たす。

主プログラム GRDNLP 計算に必要な入力データを読み、方向 \mathbf{r} や距離 λ を計算するために必要な情報量をセットし、各副プログラムを制御しながら \mathbf{r} 、 λ を計算して、収束の判定を行なう。

関数副プログラム OBJF 目的関数 $f(\mathbf{X})$ の値を計算する。

関数副プログラム ATLDF 目的関数 $f(\mathbf{X})$ の変数 $\mathbf{X} = (\mathbf{x}; \gamma)$ に関する1階の偏導関数の値を計算する。

$$\nabla f(\mathbf{X}) = \begin{cases} \left(\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_m} \frac{\partial f}{\partial \gamma_1} \dots \frac{\partial f}{\partial \gamma_n} \right) = (0 \dots 0 -1 \dots -1) & (l_1) \\ " & = (0 \dots 0 -2\gamma_1 \dots -2\gamma_n) & (l_2) \\ \left(\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_m} \frac{\partial f}{\partial \gamma} \right) = (0 \dots 0 -1) & (l_\infty) \end{cases}$$

関数副プログラム GFO 近似関数 $g(\mathbf{x}; t)$ の値を計算する。

関数副プログラム GF

$$G_j(\mathbf{X}) = \begin{cases} Cg(\mathbf{x}; t_j) - \gamma_j & (l_1 \text{ or } l_2) \\ Cg(\mathbf{x}; t_j) - \gamma & (l_\infty) \end{cases}$$

の値を計算する。

関数副プログラム ATLDGF $G_j(\mathbf{X})$ の各変数に関する1階の偏導関数の値を計算する。

$$\nabla G_j(\mathbf{X}) = \left(\frac{\partial G_j}{\partial x_1} \dots \frac{\partial G_j}{\partial x_m} \frac{\partial G_j}{\partial \gamma_1} \dots \frac{\partial G_j}{\partial \gamma_n} \right)$$

$$= \left(C \frac{\partial g(\mathbf{x}; t_j)}{\partial x_1} \dots C \frac{\partial g(\mathbf{x}; t_j)}{\partial x_m}, -\mathbf{e}_j \right) \quad (l_1, l_2)$$

但し $\mathbf{e}_j \in E^n$ は j -成分が1であるような単位行ベクトルとする。

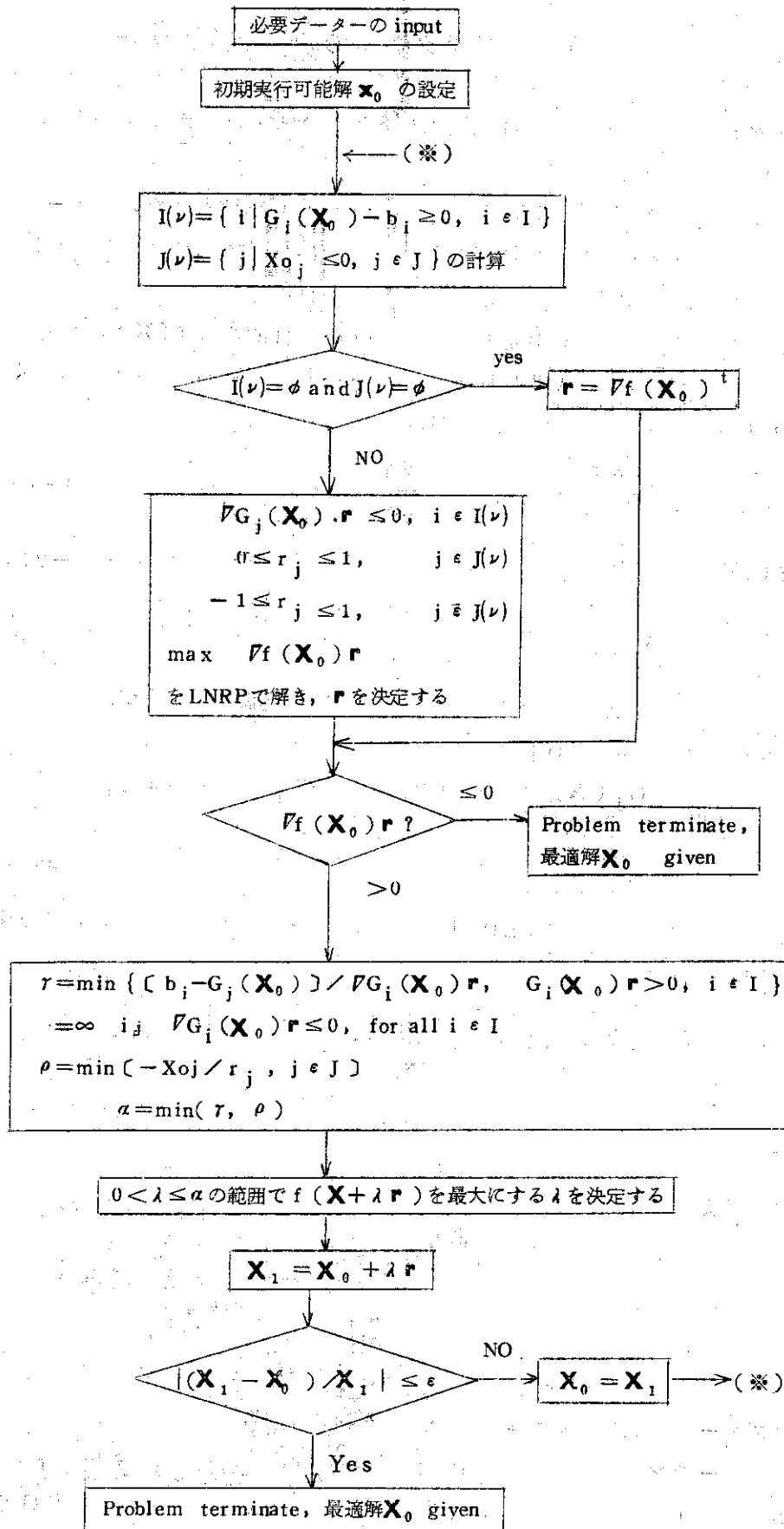
$$\nabla G_j(\mathbf{X}) = \left(\frac{\partial G_j}{\partial x_1} \dots \frac{\partial G_j}{\partial x_m} \frac{\partial G_j}{\partial \gamma} \right) \\ = \left(C \frac{\partial g(\mathbf{x}; t_j)}{\partial x_1} \dots C \frac{\partial g(\mathbf{x}; t_j)}{\partial x_m}, -1 \right) \quad (l_\infty)$$

サブルーチン副プログラム LNRP Simplex法により方向 \mathbf{r} を計算する。

サブルーチン副プログラム PRINT 必要な情報量を出力する。

これらのプログラムのうちuserが用意しなければならないのはGFOとATLDGFである。

次に主プログラムの流れ図を書いてみると



次に1つのジョブを計算するのに必要なデータを説明する。

- (1) CMNT (20A4)
 CMNT ; ジョブに対するコメント文
- (2) NORM, NMAX, MMAX, ICOND, JMAX, CONV (5I3, E7.1)
 NORM ; = 1 (l_1 ノルム), = 2 (l_2 ノルム), = 3 (l_∞ ノルム)
 NMAX ; row データの数 (≤ 100)
 MMAX ; パラメータの数 (≤ 20)
 ICOND ; = 0 (条件を付加しない), = 1 (単調性の条件を付加する),
 = 2 (凹凸性の条件を付加する)
 JMAX ; パラメータに正なる条件を付ける個数
 CONV ; 収束の判定基準 ϵ 。
- (3) T(I), I = 1, NMAX (6E12.5)
 (4) Y(I), I = 1, NMAX (6E12.5)
 T(I) ; 座標 t_i の値
 Y(I) ; 点 t_i における測定値
- (5) X(I), I = 1, MMAX (6E12.5)
 X(I) ; パラメータの初期推定値
 JMAX = 0 の時は次の(6)は不必要
 JMAX > 0 の時
- (6) JSET(J), J = 1, JMAX (24I3)
 JSET(J) ; $x_j \geq 0, j \in J$ なる条件を付けるインデックスの集合
 ICOND = 0 の時は次の(7), (8), (9)は不必要。
 ICOND = 0 の時
- (7) FX, KMAX (F6.1, I3)
 FX ; = -1.0 ($\partial g / \partial t$, or $\partial^2 g / \partial t^2 \leq b$)
 = 1.0 ($\partial g / \partial t$, or $\partial^2 g / \partial t^2 \geq a$)
 KMAX ; 条件を付加する点の数, = NMAX なら全点において条件が付加されるので
 次の(8)は不必要
- (8) K(I), I = 1, KMAX (24I3)
 K(I) ; 条件を付加する座標 t_i のインデックス
- (9) BOUND(I), I = 1, KMAX (6E12.5)
 BOUND(I) ; 条件を付加する各点における上限, または下限の値
- (10) MHD (I3)
 MHD ; 問題がパラメーターに関し線形であるか否かの情報を与える。
 = 1 (非線形)
 = 2 (線形)

4 使用例

例1 $g(x; t) = x_2 / (1 + \sqrt{1 + x_1/t}) + x_3 \sqrt{1 + x_1/t} + x_4 \quad (4)$

条件 $x_1 \geq 0$, 即ち $JMAX = 1$, $JSET(1) = 1$

例1の入力データリスト

INPUT DATA FORM II

PAGE OF

氏名	日付	プログラム名	JOB NO.	カード色指定	PUNCH
					73-80
所属	電話	研究テーマ番号	IBJOB DECKNAME	備考	YES <input type="checkbox"/> NO <input type="checkbox"/>
$G(X, T) = X(2) / (1 + \sqrt{1 + X(1)/T}) + X(3) * \sqrt{1 + X(1)/T} + X(4)$					
1	2	6	4	0	1
0.11842E+01	0.95898E+00	0.73371E+00	0.50882E+00	0.19851E+01	0.16075E+01
0.12299E+01	0.85290E+00	0.23536E+01	0.16322E+01	0.67737E+00	0.54852E+00
0.12692E+01	0.10278E+01	0.78636E+00	0.54533E+00	0.21132E+01	0.16168E+01
0.11212E+01	0.83305E+00	0.67458E+00	0.51612E+00	0.18846E+00	0.15261E+01
0.11676E+01	0.80972E+00				
0.32509E+03	0.31371E+03	0.29744E+03	0.27236E+03	0.34752E+03	0.33937E+03
0.32713E+03	0.30696E+03	0.35337E+03	0.34012E+03	0.28817E+03	0.27382E+03
0.32406E+03	0.31326E+03	0.29765E+03	0.27339E+03	0.34495E+03	0.33505E+03
0.31806E+03	0.30084E+03	0.28766E+03	0.26946E+03	0.33977E+03	0.33165E+03
0.31941E+03	0.29917E+03				
0.148E+01	0.806E+03	-0.2E+01	0.3E+00		
X(1)の初期値 X(1) ≥ 0 の場合 MHD: 非線形					

$G(X, T) = X(2) / (1 + \sqrt{1 + X(1)/T}) + X(3) * \sqrt{1 + X(1)/T} + X(4)$

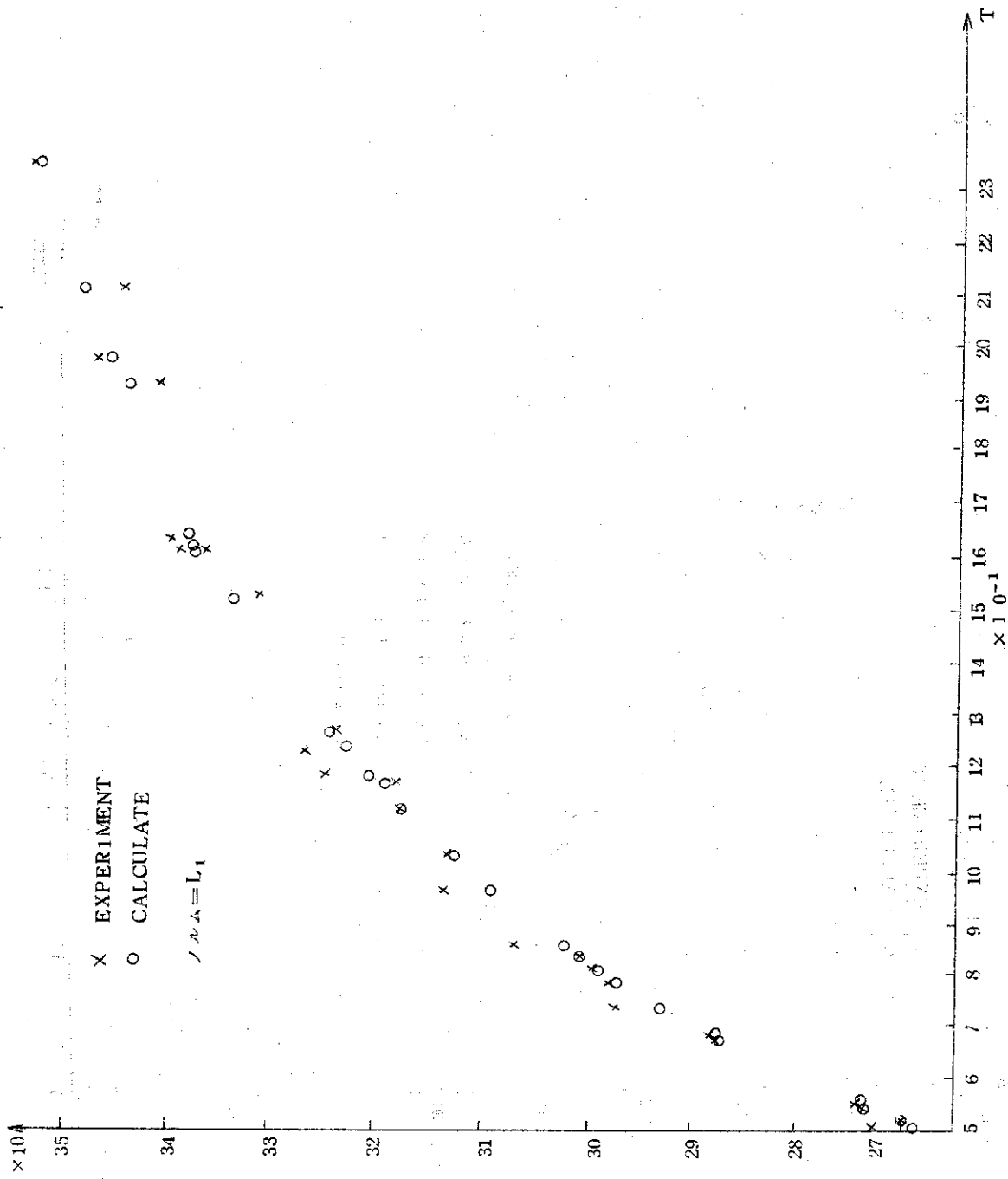
NC.	T(I)	EXPERIMENT	CALCULATE	ERROR
1	0.118420E 01	0.325090E 03	0.321100E 03	0.399038E 01
2	0.958980E 00	0.313710E 03	0.309157E 03	0.455272E 01
3	0.733710E 00	0.297440E 03	0.292789E 03	0.465135E 01
4	0.508820E 00	0.272360E 03	0.268673E 03	0.368680E 01
5	0.198510E 01	0.347520E 03	0.346240E 03	0.128024E 01
6	0.160750E 01	0.336708E 03	0.336708E 03	0.266238E 01
7	0.122990E 01	0.327130E 03	0.323145E 03	0.398458E 01
8	0.852900E 00	0.353370E 03	0.353179E 03	0.481432E 01
9	0.235360E 01	0.353370E 03	0.353179E 03	0.190895E 00
10	0.163220E 01	0.340120E 03	0.337431E 03	0.268890E 01
11	0.677370E 00	0.288170E 03	0.287676E 03	0.494301E 00
12	0.548520E 00	0.273820E 03	0.273755E 03	0.651474E 01
13	0.126920E 01	0.324060E 03	0.324821E 03	-0.761299E 00
14	0.102780E 01	0.313260E 03	0.313180E 03	0.801315E 01
15	0.786360E 00	0.297650E 03	0.297143E 03	0.506676E 00
16	0.545330E 00	0.273390E 03	0.273362E 03	0.275650E 01
17	0.211320E 01	0.344950E 03	0.348866E 03	-0.391624E 01
18	0.161680E 01	0.335050E 03	0.336982E 03	-0.193194E 01
19	0.112120E 01	0.318060E 03	0.318092E 03	-0.322647E 01
20	0.833050E 00	0.300840E 03	0.300707E 03	0.132843E 00
21	0.674580E 00	0.287660E 03	0.287409E 03	0.250992E 00
22	0.516120E 00	0.269460E 03	0.269641E 03	-0.181076E 00
23	0.188460E 01	0.339770E 03	0.343989E 03	-0.421855E 01
24	0.152610E 01	0.331650E 03	0.334202E 03	-0.255199E 01
25	0.116760E 01	0.319410E 03	0.320329E 03	-0.919052E 00
26	0.809720E 00	0.299170E 03	0.298959E 03	0.210854E 00

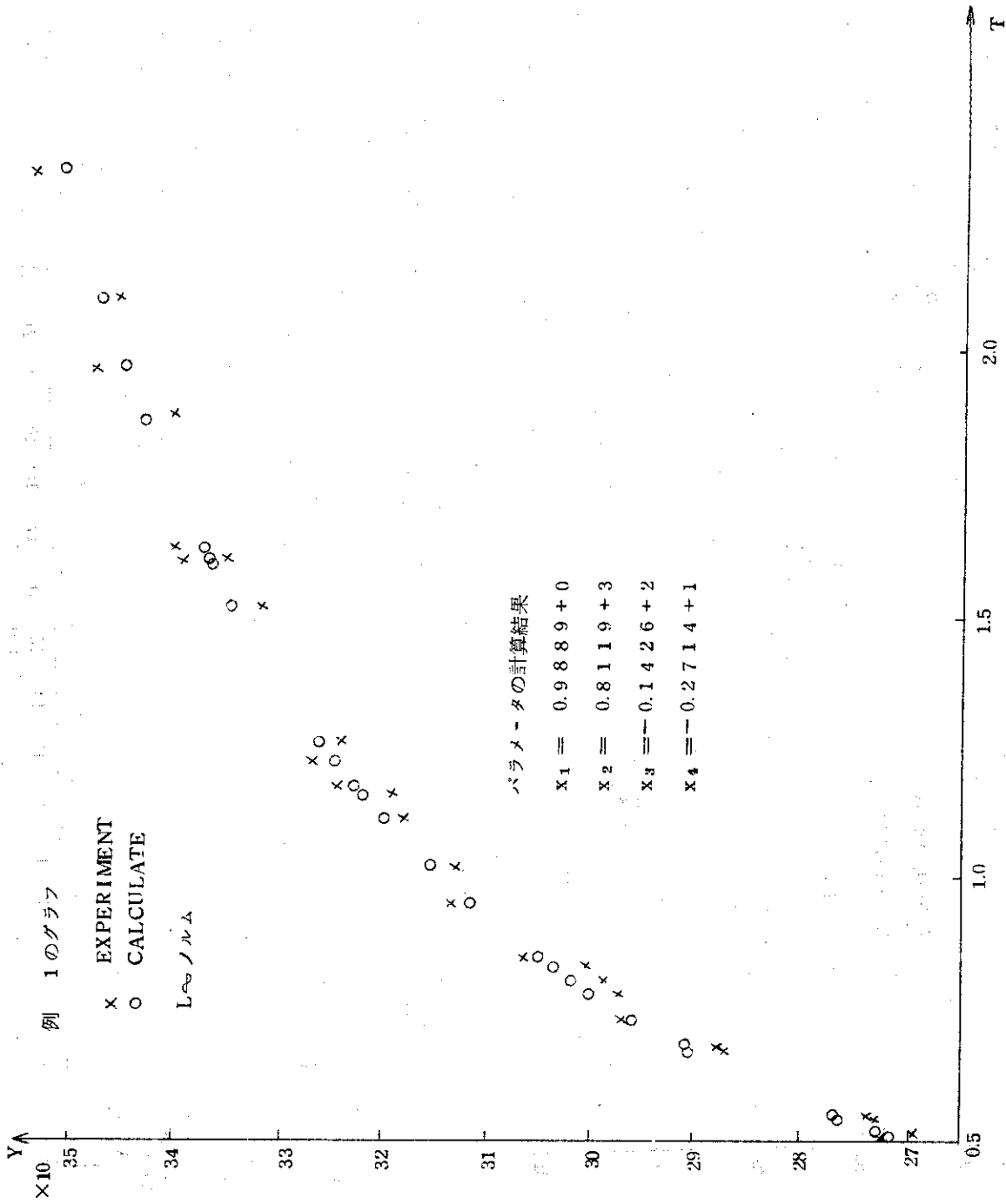
ITERATION NC. = 12

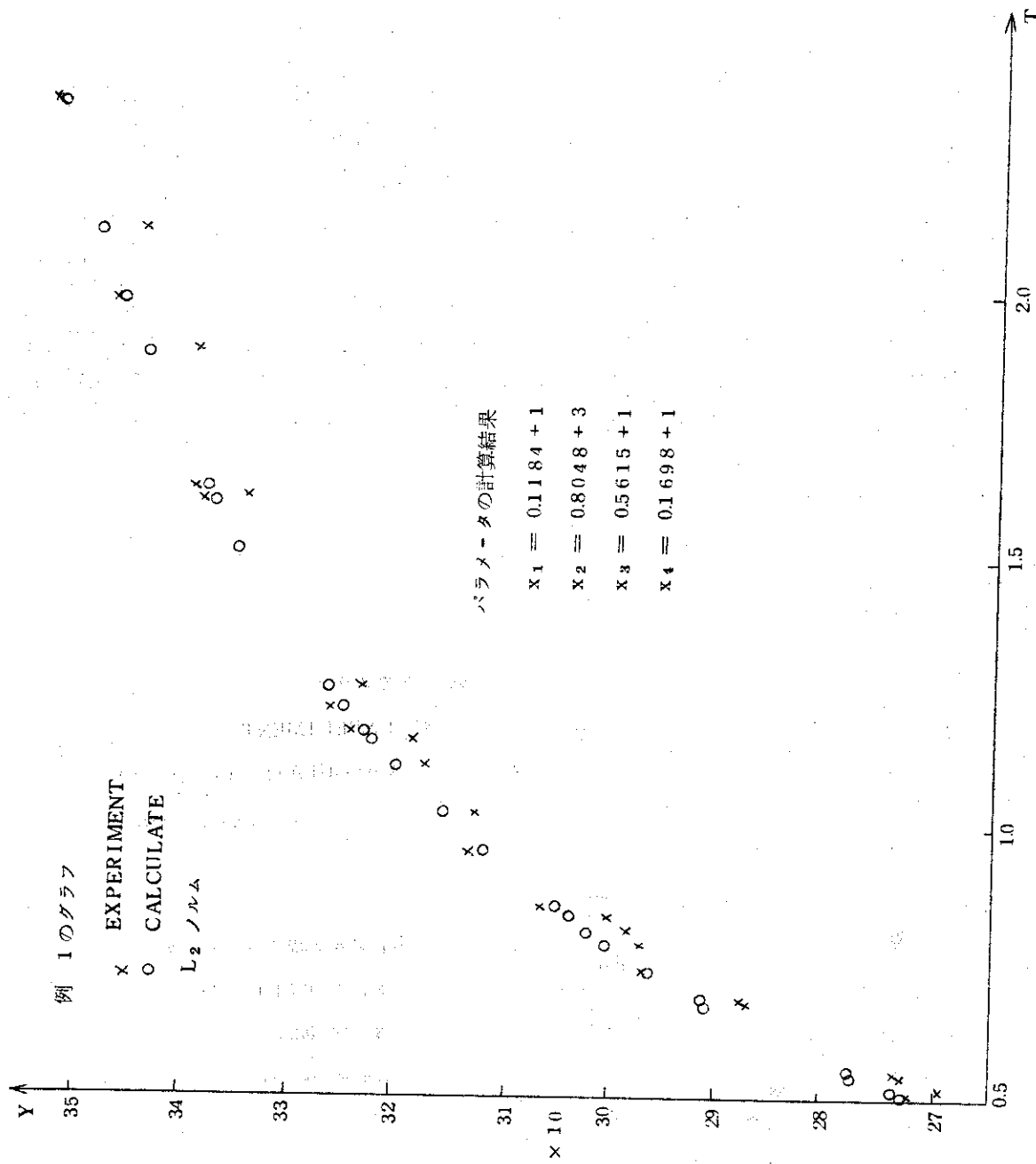
PARAMETER X(I)

0.14880E 01 0.80534E 03 -0.15203E 01 0.15272E 01

NORM = 1 CRITERION OF CONVERGENCE = 0.10E-09







$$\text{例 2 } g(x; t) = [1 / \{x_1 + x_2 (\log x_3 + t)^2\}] \times (1 + e^{-t})^{0.875} \quad [4]$$

条件 $x_3 \geq 0$, 即ち JMAX=1, JSET(1)=3

例 2 の出力リスト

$$G(X, T) = 1. / (X(1) + X(2) * ((ALOG(X(3)) + T) ** 2)) + 1. * ((1. + EXP(-T)) ** (0.875))$$

NO.	T(I)	EXPERIMENT	CALCULATE	ERROR
1	-0.460517E 01	0.128000E 03	0.126228E 03	0.177215E 01
2	-0.391202E 01	0.684000E 02	0.711406E 02	-0.274059E 01
3	-0.321888E 01	0.385000E 02	0.398111E 02	-0.131113E 01
4	-0.265926E 01	0.244000E 02	0.248920E 02	-0.491993E 00
5	-0.230259E 01	0.185000E 02	0.185056E 02	-0.561237E-02
6	-0.160944E 01	0.112000E 02	0.105910E 02	0.609001E 00
7	-0.916291E 00	0.705000E 01	0.635233E 01	0.697666E 00
8	-0.356675E 00	0.504000E 01	0.444172E 01	0.598278E 00
9	0.0	0.420000E 01	0.365142E 01	0.548585E 00
10	0.693147E 00	0.310000E 01	0.269323E 01	0.406770E 00
11	0.138629E 01	0.238000E 01	0.217753E 01	0.202467E 00
12	0.194591E 01	0.196000E 01	0.193120E 01	0.287989E-01
13	0.230259E 01	0.175000E 01	0.182045E 01	-0.704481E-01
14	0.299573E 01	0.148000E 01	0.166726E 01	-0.187262E 00
15	0.368888E 01	0.129000E 01	0.156318E 01	-0.273183E 00
16	0.424850E 01	0.117000E 01	0.149994E 01	-0.329936E 00
17	0.460517E 01	0.111000E 01	0.146609E 01	-0.356090E 00

ITERATION NO. = 19

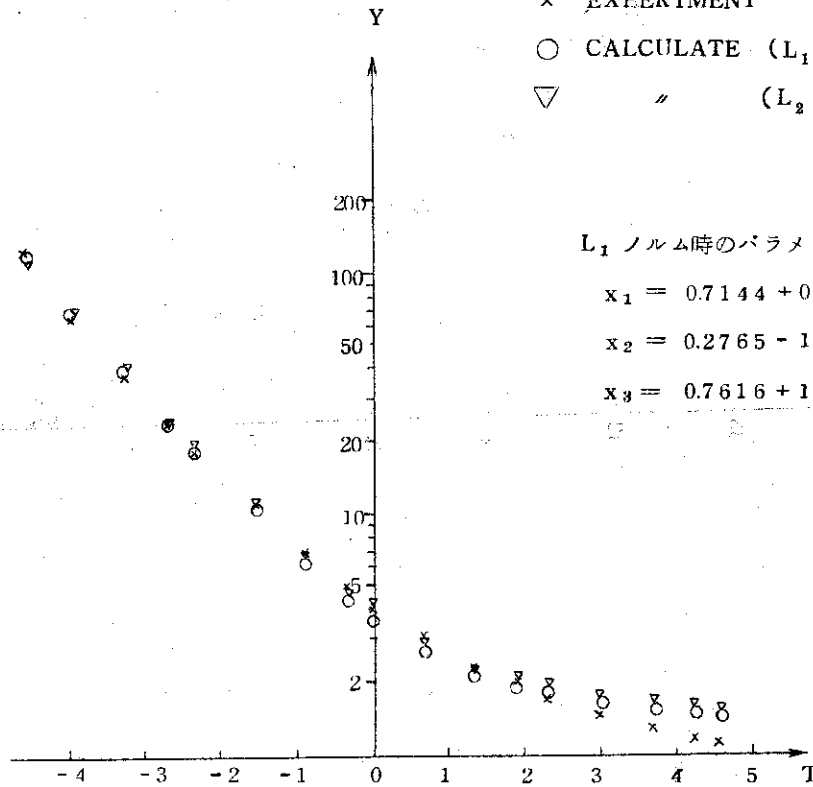
PARAMETER X(I)

0.76866E 00 0.23660E-01 0.24241E 02

NORM = 2 CRITERION OF CONVERGENCE = 0.10E-09

例 2 のグラフ

- × EXPERIMENT
- CALCULATE (L₁ ノルム)
- ▽ " (L₂ ノルム)



L₁ ノルム時のパラメータ

$$x_1 = 0.7144 + 0$$

$$x_2 = 0.2765 - 1$$

$$x_3 = 0.7616 + 1$$

例3 $g(\mathbf{x}; t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3 + x_5 t^4 + x_6 t^5$ [3]

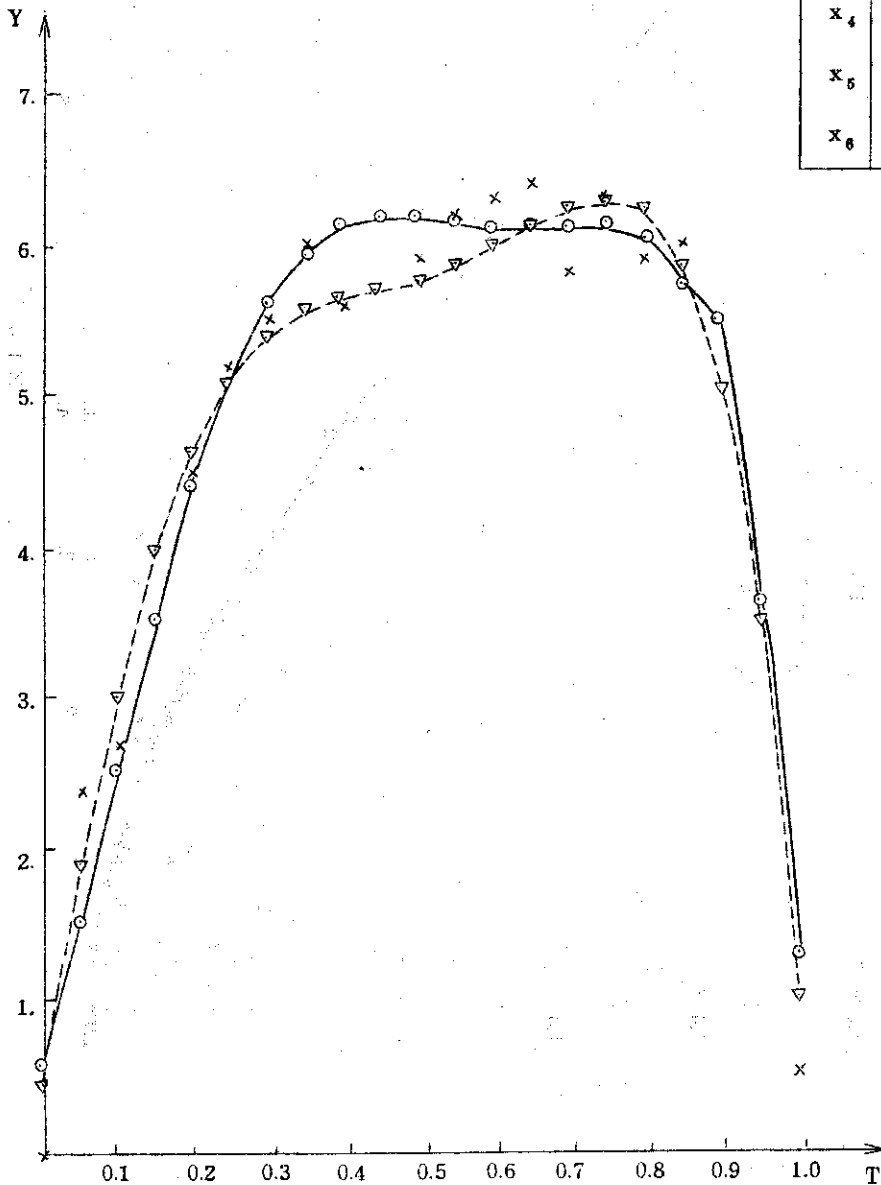
例3のグラフ

ノルム = L_∞

- × EXPERIMENT
- CALCULATE (GRADIENT法)
- ▽ " (SIMPLEX法)

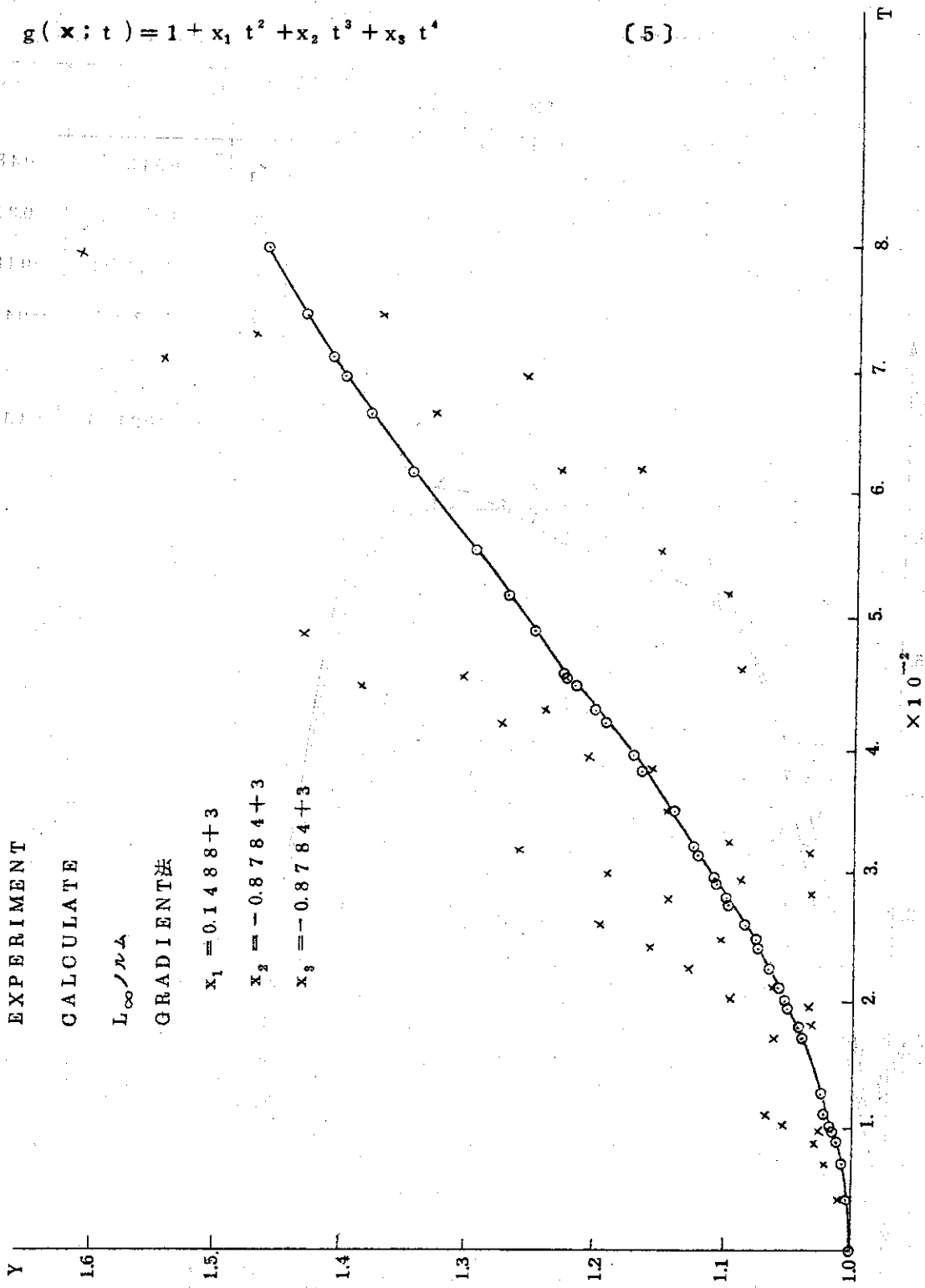
パラメータの比較

	GRADIENT 法	SIMPLEX 法
x_1	0.6513+0	0.4863+0
x_2	0.1468+2	0.3107+2
x_3	0.6873+2	-0.4873+2
x_4	-0.3222+3	-0.4176+2
x_5	0.4416+3	0.1692+3
x_6	-0.2021+3	-0.1092+3



例4 $g(x; t) = 1 + x_1 t^2 + x_2 t^3 + x_3 t^4$ [5]

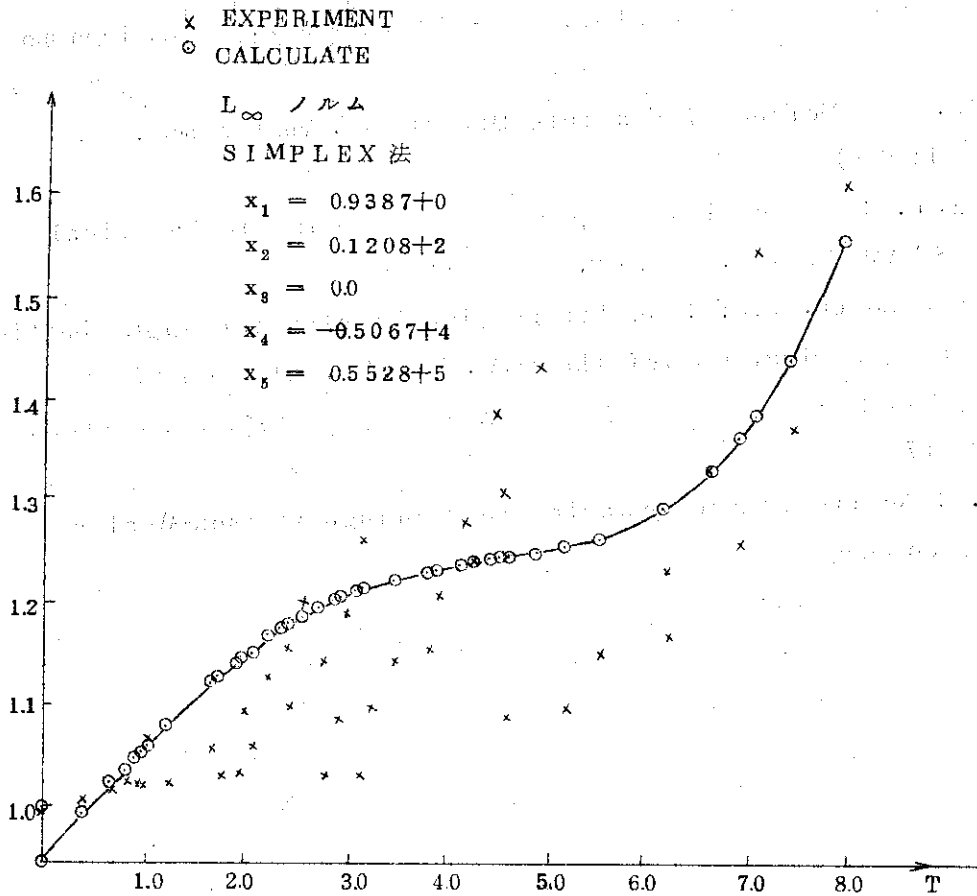
例4のグラフ



例4をSimplex法を用いて解いた場合。

但し、Simplex法を用いたプログラムLP-1では多項式近似に対しては $g(x;t) = x_1 + x_2 t + \dots + x_n t^{n-1}$ とするようになっているため、パラメーの数を5ケとして計算した。

例4のグラフ



計算時間は、例1の場合 l_1 ノルムで390秒、 l_2 ノルムで238秒、 l_∞ ノルムで20秒、例2の場合 l_1 ノルムで320秒、 l_2 ノルムで381秒、例3の場合、Simplex法で2秒、Gradient法で10秒、例4の場合Simplex法で2秒、Gradient法で9秒であった。

5 おわりに

このプログラムは G. HADLEY [6] を参考に作成したものである。プログラムでは l_1 , l_2 , l_∞ の3つのノルムの中から, user の選択により1つを選べるようにしてあるが, 他のノルムを使用する場合は目的関数 $f(\mathbf{X})$ とその偏導関数の副プログラムを作り直すことにより可能である。

プログラムのテストにさいし, テストデータを提供していただいた計算センター堀上邦彦氏に感謝いたします。

6 参考文献

- [1] 鈴木忠和: リニア・プログラミングによるカーブ・フィッティング, JAERI-memo 4542
- [2] Zoutendijk, G. : Method of Feasible Direction, Amsterdam : Elsevier (1960)
- [3] Rabinowitz, P. : Applications of Linear Programming to Numerical Analysis, SIAM Review, Vol.10, No.2 (1968)
- [4] Lafat, P. and Rosen, J.B. : An Interactive Display for Approximation by Linear Programming, Com. of the ACM, Vol.13, No.11 (1970)
- [5] 堀上邦彦, 辻本 巖 : 任意初等関数の発生方法およびその応用プログラム: LSQR, JAERI-M 4703
- [6] Hadley, G. : Nonlinear and Dynamic Programming, Addison-Wesley Publishing Company

5 おわりに

このプログラムは G. HADLEY [6] を参考に作成したものである。プログラムでは l_1 , l_2 , l_∞ の3つのノルムの中から, user の選択により1つを選べるようにしてあるが, 他のノルムを使用する場合は目的関数 $f(\mathbf{X})$ とその偏導関数の副プログラムを作り直すことにより可能である。

プログラムのテストにさいし, テストデータを提供していただいた計算センター堀上邦彦氏に感謝いたします。

6 参考文献

- [1] 鈴木忠和：リニア・プログラミングによるカーブ・フィッティング, JAERI-memo 4542
- [2] Zoutendijk, G. : Method of Feasible Direction, Amsterdam : Elsevier (1960)
- [3] Rabinowitz, P. : Applications of Linear Programming to Numerical Analysis, SIAM Review, Vol.10, No.2 (1968)
- [4] Lafat, P. and Rosen, J.B. : An Interactive Display for Approximation by Linear Programming, Com. of the ACM, Vol.13, No.11 (1970)
- [5] 堀上邦彦, 辻本 巖 : 任意初等関数の発生方法およびその応用プログラム : LSQR, JAERI-M 4703
- [6] Hadley, G. : Nonlinear and Dynamic Programming, Addison-Wesley Publishing Company

正 誤 表

頁	行	誤	正
1	I 3	初等群作不能至	初等群何種變形
1	I 7	20方型	20方達 VII
4	I 4	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1 亦 OK
		有 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 之次列法	以下仍用 λ 代 λ_1 亦 OK
		$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$
		此係變形群之變形	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 變形 A
		此係變形群	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$
5	II 2	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
6	I 4	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
7	II 4	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$	$\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1

初等群作不能至
 20方型
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
 有 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 之次列法
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$
 此係變形群之變形
 此係變形群
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$ 以下仍用 λ 代 λ_1
 $\lambda < \lambda_1 \leq \lambda_2 \leq \lambda_3$