

## 資料

## 最近の内挿法のアルゴリズムと計算プログラム\*

伊勢 武治\*\* 藤村 統一郎\*\*

## Abstract

An extensive survey has been performed on algorithms of interpolation developed during the last ten years from the viewpoint of the scientific subroutine library (SSL). The present report introduces some of new useful algorithms found from the survey. Neville's iteration scheme has been applied to achieve more precise interpolations and some algorithms have succeeded in eliminating the unnatural wiggles which had been one of the unsolved problems.

These algorithms will give a help to enlarge SSL, because they have clear mathematical bases and many of them programmed as computer subroutines.

## 1. はじめに

現在、日本原子力研究所では、科学技術計算用サブルーチンライブラリー (SSL) の拡充を計画している。これに伴い、ここ 10 年間程の数値解析関係の雑誌、AEC リポートなどのサーベイを行い、最近の内挿法のアルゴリズム、プログラムの現状などについて調べる機会を得た。その結果いくつかの新しい有用なアルゴリズムが得られたので、ここに紹介したい。

内挿法のアルゴリズムについては、多項式系 (Lagrange, Chebyshev など)、あるいは Newton の反復形 (Newton 法, Neville-Aitken 法など) などは教科書にも紹介されており、計算プログラムも公開されている<sup>1)</sup>。また、データ点のとり方によっては、異常振動 (Lagrange 内挿における Runge の問題, Chebyshev 内挿における Gibbs の現象) が起きることがわかっている<sup>2)</sup>。これらの欠点を克服するものとして、その滑らかさからスプライン関数が最近よく用いられるようになってきた<sup>3)</sup>。しかしながら、調査の結果スプライン関数にも異常振動を起こすことがあり、さらに、これを消す工夫をした内挿アルゴリズムが見つげられた。また、反復形の内挿アルゴリズムを拡張した精度の高い方法も見つけられた。これらのアルゴリズムは明快であり、また多くのものがプログラム化されてい

るので、SSL の拡充に際し有用であろう。したがって、以下に、これらのアルゴリズムの概略を紹介する。なお、普通の教科書<sup>1), 2)</sup>に書かれているようなアルゴリズムは、断りなしで用いる。

## 2. 最近のアルゴリズムの発展と計算プログラム

Lagrange 内挿は、高次で用いると Runge の現象などを起こし精度が落ちるので、低次で用いるべきであるが、比較的高次に向いている式が Aitken-Neville 内挿および重心 (barycentric) 内挿<sup>4)</sup>である。特に後者の方が向いている。これは Lagrange 内挿の変形というべきもので次式で示される。

$$f(x) = \frac{\sum_{k=0}^n \frac{A_k y_k}{x - x_k}}{\sum_{k=0}^n \frac{A_k}{x - x_k}} \quad (1)$$

ここで、

$$A_k = \frac{1}{\prod_{i=0(i \neq k)}^n (x_k - x_i)}, \quad y_k = f(x_k) \quad (2)$$

Winrich<sup>5)</sup> は、これら 3 つの内挿式の演算回数と比較を行っている。特徴のある計算機 (除算が早いとか) を使うときは、特に参考になる。

有理関数 (Padé 形, 連分数) を内挿関数とする方法は、多項式近似に比べて良い近似式が求めにくかったり、手間がかかったりする欠点を持つが、一方精度が

\* Recent algorithms and computer programs for interpolation by Takeharu ISE and Toichiro FUJIMURA (Tokai Research Establishment, Japan Atomic Energy Research Institute).

\*\* 日本原子力研究所東海研究所原子炉工学部

1桁から数桁高いので、精度が問題になるようなときは有用である。特に、極の近くでの問題 ( $e^x, \tan x$  など) に有用である。

Stoer<sup>6)</sup> は連分数に Newton 法を応用したアルゴリズムを導いている。今、連分数を次式 (Stieltjes の表現) で示す。

$$g(x) = a_0 p_0(x) + a_1 p_1(x) + \dots + a_k p_k(x) \\ + \frac{a_{k+1} p_{k+1}(x)}{1} + \frac{x - x_{k+1}}{a_{k+2}} + \frac{x - x_{k+2}}{a_{k+3}} + \dots \quad (3)$$

ここに、

$$p_0(x) = 1, p_{k+1}(x) = (x - x_k) p_k(x) \quad (k \geq 0) \quad (4)$$

ここで

$$P_l(x) = \sum_{j=0}^l a_j p_j(x), \quad (l = k, k+1) \quad (5)$$

とおくと、与えられたデータ点との関係式

$$P_{k+1}(x_i) = f(x_i), \quad i = 0, 1, \dots, k+1$$

から、係数  $a_0, a_1, \dots, a_{k+1}$  が Newton の差分商を用いて求めることができ、残りの係数は、次に示すように、逆差分を用いて求めることができる。すなわち、

$$b_{i,-1} = 0, b_{i,0} = \frac{a_{k+1} p_{k+1}(x_{k+1+i})}{f(x_{k+1+i}) - P_k(x_{k+1+i})}, \quad i = 0, 1, \dots \quad (7)$$

ここで、 $b_{i,j}$  ( $i = 1, 2, \dots$  および  $j = 1, 2, \dots, i$ ) は次の漸化式から求められる。

$$b_{i,j} = b_{i-1,j-2} + \frac{x_{k+1-i} - x_{k+1+i-j}}{b_{i,j-1} - b_{i-1,j-1}} \quad (8)$$

これらを用いて、最後に

$$\left. \begin{aligned} a_{k+2} &= b_{1,1} \\ a_{k+i} &= b_{i-1,i-1} - b_{i-3,i-3}, \quad i = 3, 4, \dots \end{aligned} \right\} \quad (9)$$

が得られる。なお、この方法で  $k=0$  とおくと連分数による Thiele の内挿法<sup>7)</sup>になる。

さらに、Larkin<sup>8)</sup> は、次式で示される有理関数  $R(x)$  に Neville 法を適用したアルゴリズムを示している。

$$R(x) = \frac{\sum_{r=0}^p a_r x^r}{\sum_{r=0}^q b_r x^r} \quad (10)$$

ここでの係数  $\{a_r; 0 \leq r \leq p\}, \{b_r; 0 \leq r \leq q\}$  は、データ点からの係数を定めるべき  $p+q+1$  の連立一次方程式からは一意的には決まらないので、反復法 (Neville-Aitken 法) で求める。問題はデータ点  $\{(x_j, f_j = f(x_j)); j = 1, 2, \dots\}$  が与えられたとき、 $k$  を反復回数とし、

$$f_{jk}(x_r) = f_r, \quad j \leq r \leq j+k$$

なる 2 次元配列  $\{f_{jk}(x); j = 1, 2, \dots; k = 1, 2, \dots\}$  を作ることにあがるが、 $\{f_{jk}\}$  を  $x$  の有理関数に限定していることから、次の、2 つの三角則 (triangle rules) と 1 つの菱形則 (rhombus rule) が得られる。

$$f_{jk} = \frac{(x - x_j) f_{j+1,k-1} + (x_{j+k} - x) f_{j,k-1}}{x_{j+k} - x_j} \quad (11)$$

$$f_{jk} = \frac{x_{j+k} - x_j}{\frac{x - x_j}{f_{j+1,k-1}} + \frac{x_{j+k} - x}{f_{j,k-1}}} \quad (12)$$

$$f_{jk} = f_{j+1,k-2} + \frac{x_{j+k} - x_j}{\frac{x - x_j}{f_{j+1,k-1} - f_{j+1,k-2}} + \frac{-x_j}{\frac{x_{j+k} - x}{f_{j,k-1} - f_{j+1,k-2}}}} \quad (13)$$

(11) 式は出発値を

$$f_{r0} = f_r, \quad j \leq r \leq j+k \quad (14)$$

としたときの Neville-Aitken 法である。したがって、最後は高々次数  $s$  の多項式  $f_{rs}$ :

$$\{f_{rs}; j \leq r \leq j+k-s, 1 \leq s \leq k\} \quad (15)$$

になる。一方、反復公式 (12) は、出発値を (14) 式とすると次の関数となる。

$$f_{rs} = \left\{ \sum_{i=0}^s a_i x^i \right\}^{-1}, \quad j \leq r \leq j+k-s, 1 \leq s \leq k \quad (16)$$

ここに、係数  $\{a_s; 0 \leq s \leq k\}$  は定数である。

〔アルゴリズム A<sub>i</sub>〕 第  $i$  列 ( $i$  回反復) までは (11) 式を用いて、それ以後は (13) 式を用いる。

〔アルゴリズム B<sub>i</sub>〕 第  $i$  列までは (12) 式を用いて、それ以後は (13) 式を用いる。

このようにして、 $\{x \rightarrow x_r; j \leq r \leq j+k\}$  まで行くと、最後に

$$f_{j,k-1}(x_{j+k}) = f_{j+1,k-2}(x_{j+k}) \quad (17)$$

あるいは、

$$f_{j+1,k-1}(x_j) = f_{j+1,k-2}(x_j) \quad (18)$$

となり、収束した求める内挿関数 (あるいは関数値) となる。Table 1 (次頁参照) は〔アルゴリズム A<sub>i</sub>〕による例で一番右側が求める関数である。

さらに、Hunter<sup>9)</sup> は、データ点が  $-\pi < x_i < \pi$  ( $i = 0, 1, \dots, n$ ) で周期性を持っているときに、内挿関数に

$$f(x) = a_0 + \sum_{k=1}^m (a_k \cos kx + b_k \sin kx) \quad (19)$$

を選び、Aitken-Neville の内挿法を適用したアルゴリズムを発表して良い結果を得ている。

Table 1 Example of a rational interpolation table by [Algorithm A<sub>i</sub>]

|     |       |                |                       |                       |                                |                     |                     |
|-----|-------|----------------|-----------------------|-----------------------|--------------------------------|---------------------|---------------------|
| $j$ | $x_j$ | $f_j$          | $k=1$                 | $2$                   | $3$                            | $4$                 | $5$                 |
| 1   | 0     | 2              | $\frac{4-2}{2}$       | $\frac{14-5x}{7-x}$   | $\frac{(2-x)(4-x)}{2(2-x)}$    | $\frac{x+2}{x^2+1}$ | $\frac{x+2}{x^2+1}$ |
| 2   | 1     | $\frac{2}{3}$  | $\frac{22-7x}{10}$    | $\frac{10-x}{2+4x}$   | $\frac{x^2-10x+48}{2(7x+6)}$   | $\frac{x+2}{x^2+1}$ |                     |
| 3   | 2     | $\frac{4}{5}$  | $\frac{14-3x}{10}$    | $\frac{22-x}{13x-1}$  | $\frac{x^2-14x+136}{2(33x+4)}$ |                     |                     |
| 4   | 3     | $\frac{1}{2}$  | $\frac{32-5x}{34}$    | $\frac{40-x}{28x-10}$ |                                |                     |                     |
| 5   | 4     | $\frac{6}{17}$ | $\frac{304-37x}{442}$ |                       |                                |                     |                     |
| 6   | 5     | $\frac{7}{26}$ |                       |                       |                                |                     |                     |

一般には Newton 内挿法は有効でないとされているが, Krogh<sup>10)</sup>は, この方法を多項式内挿および数値微分に適用して, Aitken-Neville 法より有効な, 以下に示すようなアルゴリズムを与えている. 今, データ点  $\{x_i, f(x_i); i=0, 1, \dots, n\}$  を通る次数  $n$  の多項式  $P_n(x)$  が, Newton 内挿式で与えられているとする.

$$P_n(x) = f[x_0] + \pi_1 f[x_0, x_1] + \pi_2 f[x_0, x_1, x_2] + \dots + \pi_n f[x_0, x_1, \dots, x_n] \quad (20)$$

ここに,

$$\pi_i = (x - x_0)(x - x_1) \dots (x - x_{i-1})$$

であり,  $f[x_0, x_1, \dots, x_i]$  は次に示す差分商を表わす.

$$\left. \begin{aligned} f[x_k] &= f(x_k) \\ f[x_0, x_k] &= \frac{f[x_0] - f[x_k]}{x_0 - x_k} \\ f[x_0, x_1, \dots, x_i, x_k] &= \frac{f[x_0, x_1, \dots, x_i] - f[x_0, x_1, \dots, x_{i-1}, x_k]}{x_i - x_k} \end{aligned} \right\} \quad i \geq 1 \quad (21)$$

ここで, 次のように記号を定義しておく.

$$\begin{aligned} V_{i,k} &= f(x_i), \quad i=0 \text{ のとき} \\ V_{i,k} &= f[x_0, x_1, \dots, x_{i-1}, x_k], \quad i=1, 2, \dots, k \text{ のとき} \\ w_k &= x - x_k \\ \pi_0 &= 1 \end{aligned}$$

次に以下の2つのアルゴリズムを示す.

[アルゴリズム 1]

$$\left. \begin{aligned} V_{0,0} &= f(x_0) \\ P_0 &= V_{0,0} \\ V_{0,k} &= f(x_k) \\ V_{i+1,k} &= \frac{V_{i,i} - V_{i,k}}{x_i - x_k}, \quad i=0, 1, \dots, k-1 \\ \pi_k &= w_{k-1} \pi_{k-1} \\ P_k(x) &= P_{k-1}(x) + \pi_k V_{k,k} \end{aligned} \right\} \quad k=1, 2, \dots, n$$

[アルゴリズム 2] 前半は [アルゴリズム 1] と同じだが, 後半が少し異なる.

$$V_{k-1, k-1} = V_{k-1, k-1} + (x - x_{k-1}) V_{k, k}, \quad k=n, n-1, \dots, 1$$

$$P_n(x) = V_{0,0}$$

[アルゴリズム 3 (数値微分)] まず [アルゴリズム 1] の  $V_{\mu, \nu}$  を  $c_\nu$  と置き換えて,

$$c_0 = P_n(x)$$

$$\left. \begin{aligned} \pi_i &= w_{k+i-1} \pi_{i-1} + \pi_i \\ c_k &= c_k + \pi_i c_{k+i} \end{aligned} \right\} \quad i=1, 2, \dots, n-k$$

$$k=1, 2, \dots, n-1$$

ここでの  $c_k$  が  $x$  の囲りで展開した内挿多項式の係数を与える. すなわち,

$$P_n(x) = \sum_{k=0}^n c_k (x-x_0)^k$$

$$\left. \frac{d^k P_n(x)}{dx^k} \right|_{x=x_0} = k! c_k$$

この反復形のアルゴリズムは, 予め誤差を減らすことができるという (Lyness, Moler). また [アルゴリズム 2] からは, Hermite 内挿を導くことができる.

ここで述べたアルゴリズムと他の方法との演算操作の数の比較を Table 2 に示す. これによると Lagrange 内挿が比較的よく, その次にここで述べたアルゴリズムが位置しており, Aitkin-Neville 内挿より良いことを示している.

Table 2 The number of arithmetic operations required for calculating  $P_n(x)$ .

| Interpolation's algorithms | Divisions           | Multiplications | Additions and subtractions |
|----------------------------|---------------------|-----------------|----------------------------|
| Lagrange                   | $n+1$               | $(n+1)(n+2)$    | $(n+1)(n+2)$               |
| Aitken-Neville             | $\frac{1}{2}n(n+1)$ | $n(n+1)$        | $(n+1)^2$                  |
| Algorithm 1                | $\frac{1}{2}n(n+1)$ | $2n$            | $(n+2)^2 + n - 1$          |
| Algorithm 2                | $\frac{1}{2}n(n+1)$ | $n$             | $(n+1)^2 + n - 1$          |

Gustafson<sup>11)</sup> は Krogh の [アルゴリズム 2] を Lagrange 内挿および Hermite 内挿を含む広い級に応用した。その数値例では、データ点を増やすと他の標準的な方法 (軸選択付き Gauss 消去法を採用している) より 3 倍から 10<sup>5</sup> 倍精度が良いことを示しており、求積法への応用も含めて Algol によるプログラムも与えている<sup>12), 13)</sup>。

3 次の自然スプライン関数<sup>3)</sup> による内挿プログラムは、よく計算センターなどでも整備されているが、Herriot<sup>14)</sup> は自然スプラインを高々 7 次まで探ってプログラム化 (Algol) している。節点 (データ点) が等間隔のときには、アルゴリズムが簡単になるので、そのプログラムも与えている。自然スプラインによる内挿については Greville の本<sup>15)</sup> で詳しいので省略する。自然スプラインを 7 次までしか採らなかったのは、次数が大きくなると差分商からの丸め誤差の混入により係数マトリックスが急速にたちが悪くなるからである。

スプライン関数 (特に 3 次スプライン関数) は、その平滑性から局所的ふるまいを正確に表わす良い内挿式として知られている (この辺の理論も Greville の本が詳しい)。しかし、Schweikert<sup>16)</sup> が、このスプライン関数によってあてはめられた曲線に異常屈曲点 (extraneous inflection points) が生じて正しい曲線を表わしていないことがあるのを指摘した。以下には、この異常性を克服したアルゴリズムを紹介する。

Cline<sup>17)</sup> は、この異常屈曲点に何か力 (tension) を働きかけて引き戻す必要を考え、この力を考慮したスプライン、すなわち応力スプライン (spline under

tension) を提唱して、Fig. 1 に示すように異常屈曲点がなくなることを証明した。図中に現われてくる  $\sigma$  は、これから述べる応力スプラインの中で、応力係数 (tensionfactor) と呼ばれるパラメータである。

今、データ点 (節点)  $\{x_i, y_i; x_i < x_{i+1}, i=1, 2, \dots, n\}$  が与えられていく、そこでの関数の 2 段微分までの連続と関数の連続

$$f(x_i) = y_i, \quad i=1, 2, \dots, n \quad (22)$$

を有する関数  $f$  を考えるのだが、更に  $f'' - \sigma^2 f$  も連続であると考え。すなわち、 $x_i \leq x \leq x_{i+1}$  に対して、

$$\begin{aligned} f''(x) - \sigma^2 f(x) = & [f''(x_i) - \sigma^2 y_i] \frac{x_{i+1} - x}{h_i} \\ & + [f''(x_{i+1}) - \sigma^2 y_{i+1}] \frac{x - x_i}{h_i}, \\ & i=1, 2, \dots, n-1 \quad (23) \end{aligned}$$

ここに、 $h_i = x_{i+1} - x_i$  である。

この (23) 式を (22) の条件を考慮して整理すると次式になる。

$$\begin{aligned} f(x) = & \frac{f''(x_i) \sinh(\sigma(x_{i+1} - x_i))}{\sigma^2 \sinh(\sigma h_i)} \\ & + \frac{f''(x_{i+1}) \sinh(\sigma(x - x_i))}{\sigma^2 \sinh(\sigma h_i)} \\ & + \left[ y_i - \frac{f''(x_i)}{\sigma^2} \right] \frac{x_{i+1} + x_i}{h_i} \\ & + \left[ y_{i+1} - \frac{f''(x_{i+1})}{\sigma^2} \right] \frac{x - x_i}{h_i} \quad (24) \end{aligned}$$

この  $f$  は  $[x_1, x_n]$  で連続であり、また  $\{f''(x_i), i=1, 2, \dots, n\}$  が一意的に求まれば (23) 式から  $f''$ ,  $f'$

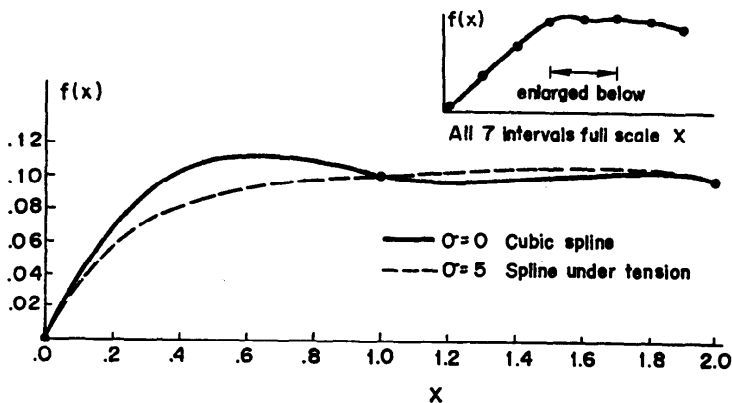


Fig. 1 Elimination of extraneous inflection points by using the spline under tension.

\* 我々が手画きで曲線あてはめを行うときには、知らずしらずのうちにこのことは考慮している。この後で出てくる Fig. 2 を見よ。

も連続であることがいえるから、次のような結果を導くことができる。

まず、(24)式から  $x_i$  の左右の微分が等しいと置くことによって次式が得られる。

$$\begin{aligned} & \left[ \frac{1}{h_{i-1}} \frac{\sigma}{\sinh(\sigma h_{i-1})} \right] \frac{f''(x_{i-1})}{\sigma^2} \\ & + \left[ \frac{\sigma \cosh(\sigma h_{i-1})}{\sinh(\sigma h_{i-1})} - \frac{1}{h_{i-1}} + \frac{\sigma \cosh(\sigma h_i)}{\sinh(\sigma h_i)} + \frac{1}{h_i} \right] \\ & \times \frac{f''(x_i)}{\sigma^2} + \frac{1}{h_i} \frac{\sigma}{\sinh(\sigma h_i)} \left] \frac{f''(x_{i+1})}{\sigma^2} \\ & = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}, \quad i=2, 3, \dots, n-1 \end{aligned} \quad (25)$$

次に、 $y_1'$  と  $y_n'$  が与えられているとして  $\{f'(x_i) = y_i'; i=1, 2, \dots, n\}$  を満足する  $f$  を考える。すなわち、 $x=x_1, x=x_n$  で(24)式を微分して等しいと置くことによって次式を得る。

$$\begin{aligned} & \left[ \frac{\sigma \cosh(\sigma h_1)}{\sinh(\sigma h_1)} - \frac{1}{h_1} \right] \frac{f''(x_1)}{\sigma^2} + \left[ \frac{1}{h_1} \frac{\sigma}{\sinh(\sigma h_1)} \right] \\ & \times \frac{f''(x_2)}{\sigma^2} = \frac{y_2 - y_1}{h_1} - y_1' \end{aligned} \quad (26)$$

$$\begin{aligned} & \left[ \frac{1}{h_{n-1}} \frac{\sigma}{\sinh(\sigma h_{n-1})} \right] \frac{f''(x_{n-1})}{\sigma^2} + \left[ \frac{\sigma \cosh(\sigma h_{n-1})}{\sinh(\sigma h_{n-1})} \right. \\ & \left. - \frac{1}{h_{n-1}} \right] \frac{f''(x_n)}{\sigma^2} = y_n' - \frac{y_n - y_{n-1}}{h_{n-1}} \end{aligned} \quad (27)$$

結局、(24)、(25)、(26)、(27)式は未知数を  $\{f''(x_i)/\sigma^2; i=1, 2, \dots, n\}$  とする3重対角マトリックスの係数行列を持つ連立1次方程式となるので、3次スプライン関数を求めるアルゴリズム(緩和法)<sup>9)</sup>で簡単に解くことができる。等高線に関連する周期関数についても、 $y_1 = f(x_{n+1}), f'(x_1) = f'(x_{n+1}), f''(x_1) = f''(x_{n+1})$  の条件を適用して、同様に解くことができる。

(23)式をみるとわかるように、 $\sigma \rightarrow 0$  とすると  $f''$  が節点で連続で、かつ節点間で1次式となるので、 $f$  は普通の3次スプライン関数となる。 $\sigma \rightarrow \infty$  とすると  $f$  それ自身に退化して、結局1次式となる。この応力スプライン関数による曲線あてはめは、飛行機・船・自動車・洋裁などのデザインに非常に有効とのことである。別の文献18)にFortranプログラムが載せられている。

Akima<sup>10)</sup>も3次スプライン内挿の欠点を克服した新しいあてはめ曲線を用いたアルゴリズムを提案している。この方法は、3次の区分多項式をいくつかのデータ点から幾何学(したがって解析的)に求めた勾配で接続していく方法で、Lagrange内挿、Fourier級数内挿などでも生ずる不自然な振動(unnatural wig-

gles)を除去することができ、熟達した製図家が画く曲線(これが最も正しい)に最も近いものが得られる。データ点の与え方が多価関数でも良いあてはめができる。

この方法は3次の区分多項式をデータ点に逐次適用して行くのであるが、各々の点での曲線の勾配を1点を中心とした5点をもって定めていく。勾配は次のようにして定める。今、5点の番号を左から1, 2, 3, 4, 5と付け、点3における勾配を求める。点3の勾配は、線分12の勾配が線23の勾配に近づいて行ったときの線分23の勾配になると考える。これより点3での勾配  $t$  は次式によって決められるとしている。

$$t = \frac{|m_4 - m_3| m_2 + |m_2 - m_1| m_3}{|m_4 - m_3| + |m_2 - m_1|} \quad (28)$$

ここに、 $m_1, m_2, m_3, m_4$  は線分12, 23, 34, 45の勾配である。したがって、端のデータ点を除いた2点間の内挿アルゴリズムは次のようになる。

2点  $(x_1, y_1), (x_2, y_2)$  の間に次の4個の条件:

$$\left. \begin{aligned} y &= y_1, \quad \frac{dy}{dx} = t_1; \quad x = x_1 \text{ で } (t_1 \text{ は勾配}) \\ y &= y_2, \quad \frac{dy}{dx} = t_2; \quad x = x_2 \text{ で } (t_2 \text{ は勾配}) \end{aligned} \right\} \quad (29)$$

があれば、一意的に3次多項式を決めることができるので、高々3次の多項式を適用するが、あてはめ曲線は次式とする。

$$y = p_0 + p_1(x - x_1) + p_2(x - x_1)^2 + p_3(x - x_1)^3 \quad (30)$$

ここに、

$$\begin{aligned} p_0 &= y_1 \\ p_1 &= t_1 \\ p_2 &= \frac{3 \frac{y_2 - y_1}{x_2 - x_1} - 2t_1 - t_2}{x_2 - x_1} \\ p_3 &= \frac{t_1 + t_2 - 2 \frac{y_2 - y_1}{x_2 - x_1}}{(x_2 - x_1)^2} \end{aligned}$$

曲線の端では勾配が決められないので、2点以上の点での値がデータ点から計算される。すなわち、端点  $(x_3, y_3)$  とその内側の2点  $(x_2, y_2), (x_1, y_1)$  を採り、次の2次式に対してあてはめられる。

$$y = g_0 + g_1(x - x_3) + g_2(x - x_3)^2, \quad (g_i \text{ は定数}) \quad (31)$$

その際、この定数  $g_i$  を求めるために、仮想的に

$$\begin{aligned} x_5 - x_3 &= x_4 - x_2 = x_3 - x_1 \\ \frac{y_5 - y_4}{x_5 - x_4} &= \frac{y_4 - y_3}{x_4 - x_3} = \frac{y_3 - y_2}{x_3 - x_2} \end{aligned}$$

$$= \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}$$

を満足する2点  $(x_4, y_4), (x_5, y_5)$  が追加される。

このように区分多項式を技巧的に決めていく Akima の方法は、他の方法と比較すると、非常に滑らかな曲線を与えることができる。Fig. 2 は、等間隔のデータ点に対して、Lagrange 内挿 (polynomial), Fourier 級数内挿, 3次スプライン内挿, osculatory 内挿<sup>24)</sup>, 等と Akima の曲線あてはめとの比較を掲げている。Akima の方法が手書きによる方法に最も近い。osculatory 内挿法によるのが次に良いが、この方法は Akima の方法によく似ているが、勾配を求めるのに5点の代りに3点を用いている。なお、図では Lagrange 法, Fourier 法共に展開項を10まで採っている。今度は、データ点の採り方を不等間隔に採って、3次スプライン法および osculatory 法と比較してみたのが Fig. 3 である。いずれの方法でも、データ点の採り方は、あてはめ曲線に重要な影響を及ぼすが、ここでの Akima の方法が、最も影響が少ないことがわかる。また、Akima の方法は、多価関数でも取り扱うことができるので、地図の等高線を画くのに適し

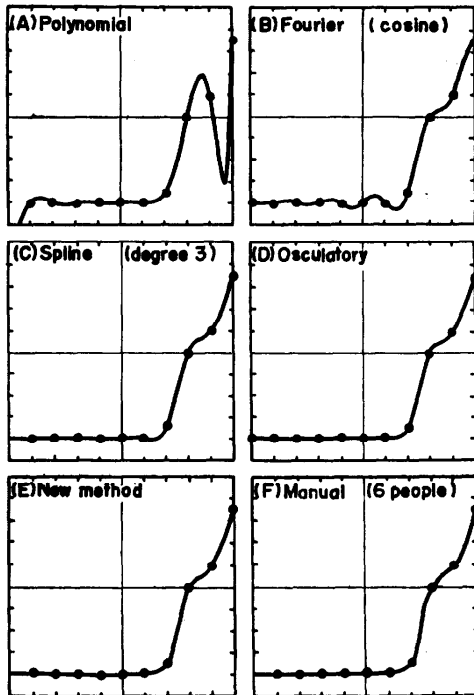


Fig. 2 Comparison of several methods of smooth curve fitting at equally spaced data points.

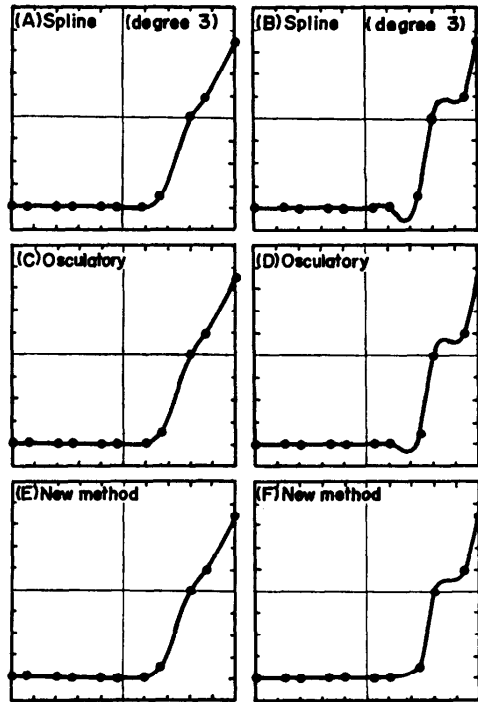


Fig. 3 Further comparison of three methods of curve fitting for sets of unequally spaced data points.

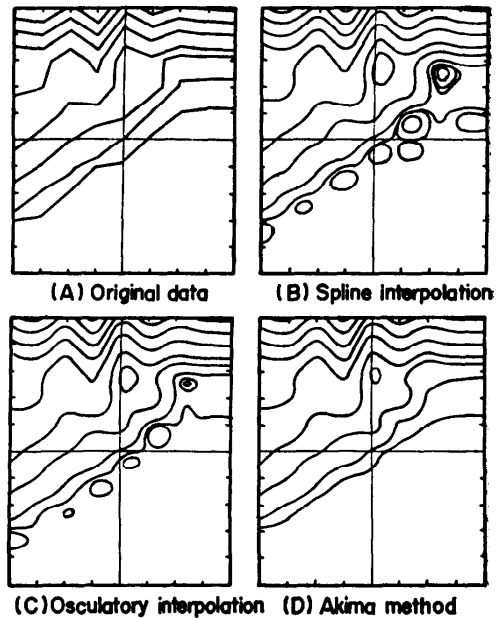


Fig. 4 Comparison of three methods for contour maps.

ている。Fig. 4 (前頁参照) にその一例を示したが、ここでも3次スプライン法および osculatory 法と異なり、非常に正確な等高線図を与えている。別の文献<sup>20)</sup>に、Fortran プログラムが載せられており、更に、 $z=f(x,y)$  型曲線に対するプログラムも記載されている<sup>22)</sup>。なお、 $z=f(x,y)$  型のときは、多価関数の取り扱いがうまくいかなかったそうである<sup>21)</sup>。

最後に、スプライン関数に似ているが、途中で人の判断が入る rounded ramp 関数 (RRF) を用いた Lagerlöf の方法<sup>23)</sup>を紹介する。彼は無限回の連続微分性を持つ、下に示すような RRF を導入した。すなわち、

$$f_{rr}(x, \alpha) = \frac{\ln(1+e^{\alpha x})}{\alpha} \quad (32)$$

この関数の性質は、

$$f_{rr}(x, \alpha) = \begin{cases} x, & \alpha x \rightarrow \infty \\ 0, & \alpha x \rightarrow -\infty \end{cases} \quad (33)$$

$$f_{rr}'(x, \alpha) = \frac{e^{\alpha x}}{1+e^{\alpha x}} \quad (34)$$

$$f_{rr}''(x, \alpha) = \frac{\alpha \cdot e^{\alpha x}}{(1+e^{\alpha x})^2} \quad (35)$$

で、 $f_{rr}''(x, \alpha)$  は  $x$  の偶関数となる。

この RRF を用いて、次の内挿近似関数を作る。

$$f(x) = \sum_{k=1}^2 \alpha_k x^{k-1} + \sum_{k=3}^N a_k f_{rr}\left(\frac{x-b_k}{c_k}, \alpha\right) \quad (36)$$

係数  $\{a_k; k=1, 2, \dots, N\}$  は、この関数の自由度であるが、この関数がデータ点  $\{x_k, y_k; k=1, 2, \dots, N\}$  を通ることから定められる。定数  $\{c_k; k=3, 4, \dots, N\}$  は規格化定数である。パラメータ  $\alpha$  は、この内挿関数の滑らかさを表わし、大きい  $\alpha$  に対しては、内挿関数はデータ点を通る折線 (直線) となる。  $b_k, c_k$  は、

$$\left. \begin{aligned} b_k &= x_{k-1} \\ c_k &= x_{k-1} - x_{k-2} \end{aligned} \right\} \quad (37)$$

から定まり、係数  $a_k$  は次のマトリックス方程式を解いて得られる。

$$\begin{pmatrix} 1, & x_1, & f_{rr}(x_{13}, \alpha), & \dots, & f_{rr}(x_{1N}, \alpha) \\ 1, & x_2, & f_{rr}(x_{23}, \alpha), & \dots, & f_{rr}(x_{2N}, \alpha) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1, & x_N, & f_{rr}(x_{N3}, \alpha), & \dots, & f_{rr}(x_{NN}, \alpha) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad (38)$$

ここに

$$x_{pq} = \frac{x_p - b_q}{c_q}$$

この係数マトリックスは、殆んど下三角形に大きな値が集中していて、上三角形は零に近い値で占められて

いる。したがって、係数  $\{a_k\}$  は反復法を用いて容易に求めることができる。パラメータ  $\alpha$  の決め方は、(36)式の2階微分をとり、その滑らかさをみて決定する。例えば、 $\sin x$  を13点分割して  $\alpha$  をパラメータにした2階微分の曲線 (Fig. 5 参照) をみると、この場合は  $\alpha=2$  が最適であることがわかる。

この RRF を用いた内挿アルゴリズムは、周期関数にも応用でき、Fourier 級数法より優れていることが

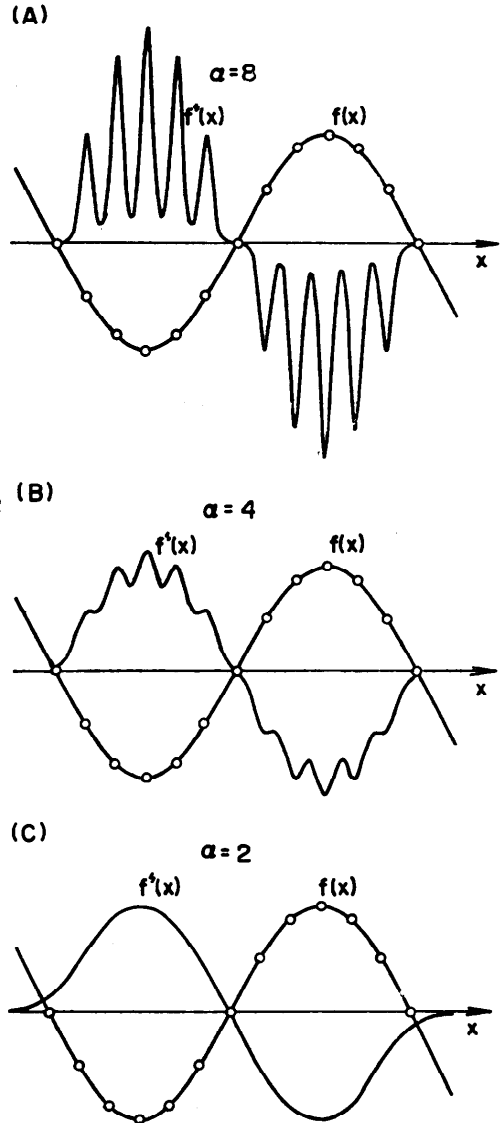


Fig. 5 Second derivative curve of  $\sin x$  interpolated through 13 points with different values of the rounding parameter  $\alpha$ .

数値的に証明されている。また、2変数以上の問題にも容易に拡張でき、この2変数の内挿公式も、この Lagerlöf 論文に示されている。末尾には、Fortran プログラムが載せられている。

### 3. あとがき

はじめに述べたように、我々の目的は当面 SSL の拡充であるので、

- (1) 現在どんなアルゴリズムが使われていて、実際の応用問題では、これらのアルゴリズムで充分解決できているのであろうか。
- (2) もし解決できてない欠点があるのであれば、その後この欠点を克服した新しいアルゴリズムの発展があるかどうか。

などに注意を払って、できるだけ広く文献をサーベイした。その結果感じたことを以下に述べる。

今まで、いろいろの計算センターで用いられているルーチンで、かなりの用が足せると思われる。しかし、極などのあるデータに対して有用な Larkin (Padé型)、Stoer (連分数) などのアルゴリズムのプログラムは加えたい。Stoer のアルゴリズムは一部の計算センターで使われているようである。Gustafson の方法も、いくつかの数値例で精度が良いようなので有用と思われる。

最近、その柔軟性からスプライン関数が内挿法にもよく用いられるようになったが、本論でも述べたようにデータ点の与え方によっては異常振動を起こすことが Schweikert によって指摘され、また、これを克服する努力が Cline, Akima 等によってなされ、その結果プログラムも完成している。これらは、いずれも有用と思われるが、我々の見たところ、現在何処の計算センターでも整備していないようである。

Table 3 には、文献サーベイの結果、プログラムリストが載っていたものをまとめておいた。これらは、

Table 3 New interpolation algorithms with program lists.

| Author (literature No.) | Language | Contents   |
|-------------------------|----------|--|
| Gustafson (11, 12, 13)  | Algol    | Extension of Newton's iterated algorithm to any functions if continuous differentiable. It contains also Lagrange and Hermite polynomial interpolations. |
| Herriot (14)            | Algol    | Natural splines up to 7th degree.  |
| Lagerlöf (23)           | Fortran  | Spline-like rounded ramp function.   |
| Cline (17, 18)          | Fortran  | Spline under tension   |
| Akima (19, 20, 21, 22)  | Fortran  | 3rd degree piece-wise polynomial, different from natural spline, with a sophisticated connection at data points.   |

いずれも有用なプログラムなので、我々としても至急整備を行いたい。また、いろいろなベンチマークテストも行いたいと考えている。ここに集録されている文献は、いずれも日本原子力研究所図書館に備えてあるので、興味を感じた方は是非原論文を参照されたい。

我々としては、できるだけ広くサーベイしたつもりであるが<sup>2)</sup>、いろいろ御気付きの点がありましたら御手数でも御連絡頂きたい。

### 参 考 文 献

(\* は Fortran が, \*\* 印は Algol プログラムが載せてある文献)

- 1)\* T.R. McCalla: Introduction to Numerical Methods and FORTRAN Programming, John Wiley (1967); 三浦功, 田尾陽一, 共訳: 数値計算法概論, 第7章 (サイエンス社).
- 2)\* 森正武: 曲線と曲面——計算機による作図と追跡, 教育出版 (1973).
- 3)\* A. Ralston & H. S. Wilf (eds.): Mathematical Methods for Digital Computers Vol. II, John Wiley (1967) Chap. 8.
- 4) R.W. Hamming: Numerical Methods for Scientists and Engineers, McGraw-Hill (1962) Chap. 8.
- 5) L.B. Winrich: Note on a comparison of evaluation schemes for the interpolating polynomial, Comp. J., Vol. 12, 154~155 (1969).
- 6) J. Stoer: A direct method for Chebyshev approximation by rational functions, JACM, Vol. 11, 59~69 (1964).
- 7) M. Abramowitz & I. A. Stegun (eds.): Handbook of Mathematical Functions, Douer (1964) Chap. 25.
- 8) F.M. Larkin: Some techniques for rational interpolation, Comp. J., Vol. 10, 178~187 (1967).
- 9) D.B. Hunter: Neville's method for trigonometric interpolation, Comp. J., Vol. 11, 311~313 (1968).
- 10) F.T. Krogh: Efficient algorithms for polynomial interpolation and numerical differentiation, Math. Comp., Vol. 24, 185~190 (1970).
- 11) S. Å. Gustafson: Rapid computation of general interpolation formula and mechanical quadrature rules, Comm. ACM, Vol. 14, 797~801 (1971).
- 12)\*\* S. Å. Gustafson: Algorithm 416, Rapid computation of coefficients of interpolation formula, Comm. ACM, Vol. 806~807 (1971).
- 13)\*\* S. Å. Gustafson: Algorithm 417, Rapid computation of weights of interpolatory quadrature rules, Comm. ACM, Vol. 14, 807 (1971).
- 14)\*\* J.C. Herriot: Algorithm 472, Procedures



- for natural spline interpolation, Comm. ACM, Vol. 16, 763~768 (1973).
- 15) T. N. E. Greville (ed.): Theory and Applications of Spline Functions, Academic (1969) の中で, p. 1~p. 36: Introduction to Spline Functions (T. N. E. Greville), p. 87~p. 102: Some Algorithms for the Computation of Interpolating and Approximating Spline Functions (L. L. Schumaker).
- 16) D. G. Schweikert: An interpolation curve using a spline in tension, J. Math. & Physics, Vol. 45, 312~317 (1966).
- 17) A. K. Cline: Scalar—and planar—valued curve fitting using splines under tension, Comm. ACM, Vol. 17, 218~220 (1974).
- 18)\* A. K. Cline: Algorithm 476, Six subprograms for curve fitting using splines under tension, Comm. ACM, Vol. 17, 220~223 (1974).
- 19) H. Akima: New method of interpolation and smooth curve fitting based on local procedures, JACM, Vol. 17, 589~602 (1970).
- 20)\* H. Akima: Algorithm 433, Interpolation and smooth curve fitting based on local procedures, Comm. ACM, Vol. 15, 914~918 (1972).
- 21) H. Akima: Method of bivariate interpolation and smooth surface fitting based on local procedures, Comm. ACM, Vol. 17, 18~20 (1974).
- 22)\* H. Akima: Algorithm 474, Bivariate interpolation and smooth surface fitting based on local procedures, Comm. ACM, Vol. 17, 26~31 (1974).
- 23)\* R. O. E. Lagerlöf: Interpolation with rounded ramp functions, Comm. ACM, Vol. 17, 476~479 (1974).
- 24) S. W. Kahng: Osculatory interpolation, Math. Comp. Vol. 23, 621~629 (1969).
- 25) 伊勢武治: 内挿法, 最小自乗法, ミニマックス近似, 科学技術計算用サブルーチンライブラリーのアルゴリズム調査, No. 11, (1975) (所内資料). (昭和50年5月19日受付)
-