スプライン関数を用いた多次元データの平滑化†

——ベクトル計算機向きの算法——

吉本 富士市†† 津田 孝夫†††

スプライン関数を用いた多次元ランダムデータの平滑化について，ベクトル計算機に適した算法を提案する．近似関数に含まれるB-スプラインは互いに独立であることに着目し，各座標軸方向ごとに，すべての標本点において集団並列的にB-スプラインの値を計算する．また，ランダムデータを節点（節線）によって作られる各部分領域ごとに分類することによって，データを同時に（並列に）扱うことを可能にする．これにより，計算量の多い部分をベクトル処理に適合させることができる．提案する算法では，3次元のランダムデータで標本点の数が節点の数よりも十分多いとき，10倍以上の加速率を得ることができる（富士通のベクトル計算機（スーパコンピュータ）VP-200 を用いた場合）．なお，多次元データの平滑化において，一意的な最小二乗解が存在するか否かを判定するため，多次元に拡張した Schoenberg-Whitney の条件（十分条件）も示している．

1. はじめに

近年，単一の多項式では近似しにくいデータの平滑化にスプライン関数を用いることが多くなってきた[1),2)]．スプライン関数は，区分的解析関数であるため，局所的（local）な振舞いをする性質がある．したがって，単一の多項式を用いたときによく見られる異常な振動（ルンゲの現象）をあまり生じない特長がある．

スプライン関数を用いたデータ平滑化においても，他の多くの問題と同様に，取り扱う問題の次元数の増加とともに計算量は指数関数的に増大する．したがって，多次元になるほど高速な計算法が望まれている．

この対策として，ベクトル計算機（スーパコンピュータ）の有効利用が考えられる．ところが，ベクトル計算機は個性の強い計算機であり，その特長を生かすような計算法およびプログラミング技術を用いなければ，その性能を十分発揮させられないことが多い[3)]．スプライン関数を用いた多次元ランダムデータの平滑化もその一例であり，ベクトル計算機の特長を考慮していない従来の計算法では，高速な計算は期待できない．このため，ベクトル計算機向きの計算法の開発が望まれているが，まだあまり研究されていないようである．

本論文では，スプライン関数を用いた多次元ランダムデータの平滑化について，ベクトル計算機に適した算法を提案する[4)]．ベクトル計算機に適合させるため，特に次の二点を考慮している．まず第一に，近似関数に含まれるB-スプラインは互いに独立であることに着目し，各座標軸方向ごとに，すべての標本点において集団並列的にB-スプラインの値を計算する．第二に，ランダムデータを節点（節線）によって作られる各部分領域ごとに分類することによって，データを同時に（並列に）扱うことを可能にする．これにより，計算量の多い部分をベクトル処理に適合させることができるようになることを示す．

なお，多次元データの平滑化において，一意的な最小二乗解が存在するか否かを判定するため，多次元に拡張した Schoenberg-Whitney の条件（十分条件）についても述べる．

2. 近似関数の表現

データ平滑化を考えている n 次元の超直方体領域を $R=[a_1, b_1]\times\cdots\times[a_n, b_n]$ とする．区間 $[a_1, b_1], \cdots, [a_n, b_n]$ をそれぞれ $t_1, \cdots, t_n$ 個に分割し，

$$\left.\begin{array}{l} a_1=P_0^{(1)}<P_1^{(1)}<\cdots<P_{t_1}^{(1)}=b_1 \\ \cdots\cdots\cdots\cdots \\ a_n=P_0^{(n)}<P_1^{(n)}<\cdots<P_{t_n}^{(n)}=b_n \end{array}\right\} \quad (1)$$

と書く．ここで，P の右肩につけたカッコ書きの数字は n 次元空間の各座標軸方向を表している．式(1)の各分点 P の上に節点を置いて，各座標軸方向のB-スプラインを作り，それらのテンソル積で近似関数を構成する*．

† Multivariate Data Fitting with a Spline Function—A Method for a Vector Computer—by FUJIICHI YOSHIMOTO (Akashi College of Technology) and TAKAO TSUDA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 明石工業高等専門学校
††† 京都大学工学部情報工学科

いま，座標軸 $x^{(q)}$ 方向のB-スプラインの階数（次数＋1）を m_q $(q=1,2,\cdots,n)$ とすると，区間 $[a_q, b_q]$ の両端では節点を m_q 個重ねて，

$$\left.\begin{aligned} a_q &= \xi_{-m_q+1}^{(q)} = \cdots = \xi_0^{(q)} \\ b_q &= \xi_{h_q-m_q+1}^{(q)} = \cdots = \xi_{h_q}^{(q)} \\ &(q=1,2,\cdots,n) \end{aligned}\right\} \quad (2)$$

とできる．内部の節点 $\xi_1^{(q)}, \xi_2^{(q)}, \cdots, \xi_{h_q-m_q}^{(q)} (q=1,2,\cdots,n)$ は，$x^{(q)}$ 方向の各分点の上で m_q 個まで重ねることができる．内部の節点を一つも重ねないで，各分点 $P_i^{(n)}$ の上に一つずつ取ったとき，近似関数は領域 R において通常のスプライン関数となる．しかし，内部の節点をどこかで重ねると拡張スプライン関数となる[2]．ここでは，この両方のスプライン関数を取り扱うことにする．節点の多重度は各分点ごとに異なっていてもよいが，ここではアルゴリズムを簡単にするため，各座標軸方向ごとに同一として，r_q と書くことにする．座標軸が異なれば，多重度 r_q（すなわち近似関数の連続性）は異なっていてもよい．

節点の多重度は，各分点での近似関数の滑らかさ（連続条件）に関係しており，r_q 重節点での連続性はその座標軸方向に対して $C^{m_q-r_q-1}$ 級（m_q-r_q-1 次微係数まで連続）となる．

さて，このように節点を決めると，平滑化を考えている領域 R において近似関数は

$$S(x^{(1)},\cdots,x^{(n)}) = \sum_{i_1=1}^{h_1} \cdots \sum_{i_n=1}^{h_n} c_{i_1,\cdots,i_n} N_{m_1,i_1}(x^{(1)}) \cdots N_{m_n,i_n}(x^{(n)}) \quad (3)$$

と表すことができる．ここで，$N_{m_q,i_q}(x^{(q)})$ は正規化された m_q 階（m_q-1 次）のB-スプラインであり，次のような局所性をもっている．

$$N_{m_q,i_q}(x^{(q)}) \begin{cases} >0 & \xi_{i-m}^{(q)} < x^{(q)} < \xi_i^{(q)} \\ =0 & \text{その他．} \end{cases} \quad (4)$$

ここで，簡単のため座標軸を表す q を省略して，$m_q \equiv m$, $i_q \equiv i$ と書くと，B-スプラインの値は，

$$M_{1,i}(x^{(q)}) = \begin{cases} (\xi_i^{(q)} - \xi_{i-1}^{(q)})^{-1} & \xi_{i-1}^{(q)} \leqq x^{(q)} < \xi_i^{(q)} \\ 0 & \text{その他} \end{cases} \quad (5)$$

を出発値として，次の漸化式

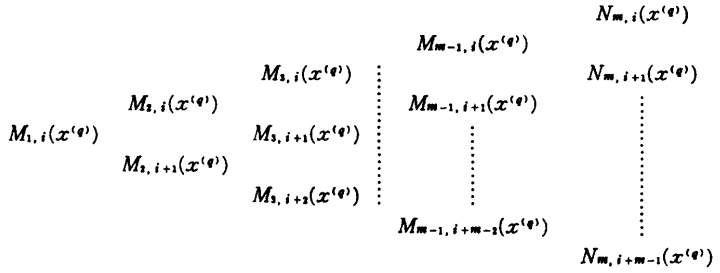

図1 B-スプラインの値の計算順序
Fig. 1 Order of computation for the values of B-splines.

$$M_{r,i}(x^{(q)}) = \{(x^{(q)} - \xi_{i-r}^{(q)}) M_{r-1,i-1}(x^{(q)}) + (\xi_i^{(q)} - x^{(q)}) M_{r-1,i}(x^{(q)})\} / (\xi_i^{(q)} - \xi_{i-r}^{(q)}) \quad (r=2,3,\cdots,m-1) \quad (6)$$

および

$$N_{m,i}(x^{(q)}) = (x^{(q)} - \xi_{i-m}^{(q)}) M_{m-1,i-1}(x^{(q)}) + (\xi_i^{(q)} - x^{(q)}) M_{m-1,i}(x^{(q)}) \quad (7)$$

を用いて容易に計算できる[4),6)]．これを，de Boor-Cox のアルゴリズムと呼ぶ．

ところで，(3)に含まれるB-スプラインは，領域 R 内の任意の点で，各座標軸方向に m_q 個の零でない値をもつが，これらの値は次のようにして計算できる．すなわち，$\xi_{i-1}^{(q)} \leqq x^{(q)} < \xi_i^{(q)}$ とすると，**図1**の配列要素の値のみを計算すればよい．ここで，$q=1,2,\cdots,n$ である．

$M_{1,i}(x^{(q)})$ から計算を始め，左から右へと計算を進める．各要素は，その左側の二つの要素（一つはすぐ上の要素，もう一つはすぐ下の要素）を用いて計算できる．たとえば，$M_{3,i+1}(x^{(q)})$ は二つの要素 $M_{2,i}(x^{(q)})$ と $M_{2,i+1}(x^{(q)})$ を用いて計算できる．この配列に表れない要素はすべて零である．したがって，たとえば $M_{3,i}(x^{(q)})$ は $M_{2,i}(x^{(q)})$ のみを用いて計算できる．

3. 最小二乗法による平滑化

データ平滑化を考えている領域 R を任意の格子により分割し，そのすべての格子点上でデータを与えるとき格子点データと呼び，そうでないときランダムデータと呼ぶ．本論文では，**ランダムデータ**の平滑化を取り扱うことにする．

n 次元データを

$$F_k = f(x_k^{(1)},\cdots,x_k^{(n)}) + \varepsilon_k \quad (k=1,2,\cdots,N) \quad (8)$$

と書く．ここで，$f(x^{(1)},\cdots,x^{(n)})$ は未知の n 変数関数（信号）であり，ε_k は測定誤差を表している．データ F_k を与える点 $(x_k^{(1)},\cdots,x_k^{(n)})$ を標本点と呼ぶ．

---

* 本論文では，分点と節点は別の概念として使っている．分点は多項式片（polynomial piece）のつなぎ目である．一方，節点はB-スプラインを定義するときの $\xi_i^{(q)}$ である（式(4)～式(7)参照）．節点は，各分点の上でいくつか重ねることができるが，重ねる数によって近似関数の連続性が異なる．

ところで，(3)にはパラメタとしてB-スプラインの係数 $c_{i_1,\cdots,i_n}$ $(i_1=1,2,\cdots,h_1;\ \cdots;\ i_n=1,2,\cdots,h_n)$ と節点 $\xi_i^{(q)}$ $(i=1,2,\cdots,h_q-m_q;\ q=1,2,\cdots,n)$ が入っており，しかもこの節点は非線形パラメタである．したがって，一般には非線形最小二乗法の問題となり，極値を複数個もつ最適化問題を解かなくてはならない．周知のように，これはむずかしい．そこで，本論文では節点は別の方法によりあらかじめ決めることができるものとして，パラメタ $c_{i_1,\cdots,i_n}$ $(i_1=1,2,\cdots,h_1;\ \cdots;\ i_n=1,2,\cdots,h_n)$ のみを未知数とする線形最小二乗法の問題を扱うことにする（節点の決定方法については5章を参照されたい）．

データ(8)の近似関数として(3)を用いる．このとき，重み付き残差の二乗和

$$Q=\sum_{k=1}^{N} w_k\{S(x_k^{(1)},\cdots,x_k^{(n)})-F_k\}^2 \qquad (9)$$

を最小にするようにパラメタ $c_{i_1,\cdots,i_n}$ $(i_1=1,2,\cdots,h_1;\ \cdots;\ i_n=1,2,\cdots,h_n)$ のみを決定すればよいことになる．ここで，w_k はデータの重みである．

式(9)をパラメタ $c_{i_1,\cdots,i_n}$ で偏微分して零とおくと，正規方程式

$$\boldsymbol{Ac}=\boldsymbol{d} \qquad (10)$$

を得る．ここで，

$$\boldsymbol{c}=(c_{1,1,\cdots,1},c_{2,1,\cdots,1},\cdots,c_{h_1,h_2,\cdots,h_n})^T \qquad (11)$$

$$\boldsymbol{d}=(d_1,d_2,\cdots,d_{h_1\ h_2\ \cdots\ h_n})^T \qquad (12)$$

である．式(10)の係数行列 A は，B-スプラインの局所性(4)から，幅が

$$2\{h_1h_2\cdots h_{n-1}(m_n-1)+h_1h_2\cdots h_{n-2}(m_{n-1}-1)+\cdots+h_1(m_2-1)+m_1\}-1$$

の対称な帯行列となる．たとえば，$n=2$, $m_1=m_2=4$（2次元の双3次スプライン関数）で

$$\begin{aligned}a_1&=\xi_{-3}^{(1)}=\cdots=\xi_0^{(1)}=P_0^{(1)}\\&<\xi_1^{(1)}=P_1^{(1)}<\xi_2^{(1)}=P_2^{(1)}\\&<\xi_3^{(1)}=P_3^{(1)}<\xi_4^{(1)}=\cdots=\xi_7^{(1)}\\&=P_4^{(1)}=b_1\end{aligned} \qquad (13)$$

$$\begin{aligned}a_2&=\xi_{-3}^{(2)}=\cdots=\xi_0^{(2)}=P_0^{(2)}\\&<\xi_1^{(2)}=P_1^{(2)}<\xi_2^{(2)}=P_2^{(2)}\\&<\xi_3^{(2)}=P_3^{(2)}<\xi_4^{(2)}=P_4^{(2)}\\&<\xi_5^{(2)}=\cdots=\xi_8^{(2)}=P_5^{(2)}=b_2\end{aligned} \qquad (14)$$

のとき，

$$A=\begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} & & & & \\
A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & & & \\
A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & & \\
A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & A_{47} & \\
 & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} & A_{57} & A_{58} \\
 & & A_{63} & A_{64} & A_{65} & A_{66} & A_{67} & A_{68} \\
 & & & A_{74} & A_{75} & A_{76} & A_{77} & A_{78} \\
 & & & & A_{85} & A_{86} & A_{87} & A_{88}
\end{bmatrix} \qquad (15)$$

$$A_{ij}=\begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & & & & \\
a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & & & \\
a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & & \\
a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & \\
 & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & \\
 & & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & \\
 & & & a_{74} & a_{75} & a_{76} & a_{77} &
\end{bmatrix} \qquad (16)$$

$$(i=1,\ 2,\ \cdots,\ 8;\ j=1,\ 2,\ \cdots,\ 8)$$

という形になる．ここで，

$$a_{uv}=\sum w_k N_{4,u}(x_k^{(1)})N_{4,i}(x_k^{(2)}) \times N_{4,v}(x_k^{(1)})N_{4,j}(x_k^{(2)}) \quad (u=1,2,\cdots,7;\ v=1,2,\cdots,7) \tag{17}$$

であり，$\sum$はB-スプラインの局所性(4)から

$$\{x_k^{(1)}\in[\xi_{u-4}^{(1)},\xi_u^{(1)}]\cap[\xi_{v-4}^{(1)},\xi_v^{(1)}]\} \cap\{x_k^{(2)}\in[\xi_{i-4}^{(2)},\xi_i^{(2)}]\cap[\xi_{j-4}^{(2)},\xi_j^{(2)}]\}$$

となる標本点 $(x_k^{(1)},x_k^{(2)})$ に対してのみ計算すればよい．なお，(15)の矩形で囲んだ16個ずつの要素が，左上から右下の順に区間 $[P_0^{(2)},P_1^{(2)}],\cdots,[P_4^{(2)},P_5^{(2)}]$ に対応している*．同様に，(16)の矩形で囲んだ16個ずつの要素が，左上から右下の順に区間 $[P_0^{(1)},P_1^{(1)}],\cdots,[P_3^{(1)},P_4^{(1)}]$ に対応している*．式(10)は，係数行列 A が対称な帯行列となるので，たとえば，修正コレスキー法を用いて効率的に解くことができる．その解 $\boldsymbol{c}$ を(3)へ代入すると，ある節点の組に対する近似関数を求めることができる．

ところで，(10)が唯一の解をもつためには，Schoenberg-Whitney の条件（以下 S-W 条件と略記する）が成り立たなければならない．すなわち，(3)の中のB-スプライン基底 $N_{m_1,i_1}(x^{(1)})\cdots N_{m_n,i_n}(x^{(n)})$ $(i_1=1,2,\cdots,h_1;\cdots;i_n=1,2,\cdots,h_n)$ に対して，標本点がそれぞれ1つ以上存在しなければならない．1次元の場合には，この条件が成り立つか否かを厳密に調べるアルゴリズムを作ることができるが，それを多次元へ拡張することは容易でない．したがって，ここでは次の式を用いて S-W 条件を調べることにする．

いま，

$$Q_q=\mathrm{NINT}((m_q+(t_q-1)r_q)/t_q)+1 \quad (q=1,2,\cdots,n) \tag{18}$$

とおき，n 次元の場合の S-W 条件の判別式を

領域 $[P_{q_1-1}^{(1)},P_{q_1}^{(1)}]\times\cdots\times[P_{q_n-1}^{(n)},P_{q_n}^{(n)}]$ のデータの数 $\geqq Q_1Q_2\cdots Q_n$

$$(q_i=1,2,\cdots,t_i;\ i=1,2,\cdots,n) \tag{19}$$

とする．ここで，NINT は“四捨五入後整数化”を表す．式(18)の右辺の $m_q+(t_q-1)r_q$ は，座標軸 $x^{(q)}$ 方向に対して，近似関数 $S(x^{(1)},\cdots,x^{(n)})$ に含まれるB-スプラインの数 h_q に等しい．ゆえに，(19)の右辺は，1領域当りのB-スプラインの数（すなわち，1領域当りの必要なデータの数）よりも少し大きな値である．式(19)は厳密な S-W 条件ではなく，その十分条件となっていることに注意したい．ここでは，(19)が満足されないとき，正規方程式(10)は一意的な解をもたないものと判定する．

多次元のランダムデータで，データに偏りがある場合には，S-W 条件を満たさないこともあるので，この条件は重要である．

4. 平滑化の評価規準

スプライン関数を用いたデータ平滑化では，S-W 条件を満たす範囲内であれば，平滑化を考えている領域 R 内の分点の数と位置（したがって，節点の数と位置）は，自由に決めることができる．しかし，分点の数が多すぎると，近似関数がデータに追従しすぎて誤差を除くことができなくなる．逆に，分点の数が少なすぎると，近似関数はデータのもとにある関数（信号）をうまく表現できなくなる．また，たとえ分点の数はよくても，その位置が悪いと良い近似を得ることは困難である．したがって，分点の数と位置をいろいろ変えて平滑化を計算し，その中から良いものを選び出す必要がある．このためには，平滑化の良さを評価する規準がなければならない．特に，3次元以上になると人間が平滑化の結果を目で見て評価することが容易でないので，客観的な評価規準があれば有益である．この規準として，ここでは誤差分散の不偏推定量 δ と赤池の情報量規準 AIC を用いることにする[2]．

n 次元の場合には，回帰模型が

$$F_k=S(x_k^{(1)},\cdots,x_k^{(n)})+\varepsilon_k \quad (k=1,2,\cdots,N) \tag{20}$$

となるので，誤差分散の不偏推定量 δ は

$$\delta=Q/(N-h_1h_2\cdots h_n) \tag{21}$$

となり，情報量規準 AIC は

$$\mathrm{AIC}=N\log_e Q+2h_1h_2\cdots h_n \tag{22}$$

と書くことができる．ここで，Q は(9)で与えられ，$h_1h_2\cdots h_n$ は近似関数 $S(x^{(1)},\cdots,x^{(n)})$ のパラメタ $c_{i_1,\cdots,i_n}$ $(i_1=1,2,\cdots,h_1;\cdots;i_n=1,2,\cdots,h_n)$ の数である．

誤差分散の不偏推定量の値は，通常は近似関数に含まれるパラメタの数の増大とともに減少していき，ある段階でほとんど一定の値となる．そこで，この一定になりかけた最初のパラメタ数の場合の平滑化を最もよいものとする．また AIC の値も，理想的には近似関数のパラメタの数の増加とともに減少していき，ある段階で最小の値となり，その後はわずかに増加する傾向を示す．したがって，この最小となるパラメタ数の場合の平滑化を最もよいものとする．ただし，分点の位置が最適でない場合には，このように“きれいな”結果にならないこともある．

* 区間の境界上の標本点は右側の区間に入るものとして扱う．

5. 分点の決定方法

スプライン関数を用いるとき，良い近似を得るための鍵は，領域の分点（したがって節点）の決定である．特に，多次元になると人間が目で見て分点の良さを評価することが難しくなるので，その自動的な決定方法が望まれる．しかし，残念ながらこの問題は難しく，まだ決定的な方法はないようである．その理由は，分点上の節点が近似関数の中に非線形パラメタの形で入っているからであろう．ここでは，少ない分点数で平滑化を始め，適応的に最も平滑化の悪い領域へ分点を追加しながら繰り返し平滑化を計算することにより，良い分点を決定する方法を用いることにする．最も平滑化の悪い領域としては，各領域において重み付き残差の二乗和が最大の領域をとる．この方法は最適な分点を与えるのではないが，簡単にまずまずの結果を得ることができる．

なお，ここで述べた方法で分点を決めた後で，その分点の位置だけを少し動かしてみて，さらに微調整を行うとよいこともある．その算法は，たとえば文献2)の最小化のアルゴリズムを参照されたい．

6. ベクトル計算機向きの算法

以上述べたことから，スプライン関数を用いた多次元ランダムデータ平滑化のアルゴリズム全体の構成は，図2のようになる．図2の各ステップの中で，特に演算量が多いところは，B-スプラインの値の計算，正規方程式の要素の計算，正規方程式の解の計算および平滑化の評価規準の計算である．したがって，まずこの部分をベクトル計算機に適合させる必要がある．一般に，ベクトル計算機向きの算法を考える場合のキーポイントは，ベクトル化率を高くすることとベクトル演算を高速化することである．ベクトル演算を高速化するためには，ベクトル長を長くすること，パイプライン演算器を有効に利用すること，などが必要である．スプライン関数を用いたデータ平滑化では，特にベクトル長をいかにして長くするかという問題が高速化の鍵となっている．

まず，B-スプラインの値の計算をベクトル計算機に適合させることを考えよう[5),6)]．このためには，2章で述べた de Boor-Cox のアルゴリズムをベクトル計算機向きにすればよい．一つ一つの標本点に対して逐次的に図1の計算を行ったのでは，階数 m_q が大きくないため（通常は $m_q \leqq 10$），ベクトル長を長くできない．ところが，近似関数(3)に含まれるB-スプライン $N_{m_q,i_q}(x^{(q)})$ $(i_q=1,2,\cdots,h_q;\ q=1,2,\cdots,n)$ は互いに独立であり，節点が決まればすべて決まってしまうものである．したがって，平滑化を考えている領域 R 内で，**各座標軸方向ごとに**，B-スプラインの値を計算したいすべての点 $x_k^{(q)}$ $(k=1,2,\cdots,N)$ におい

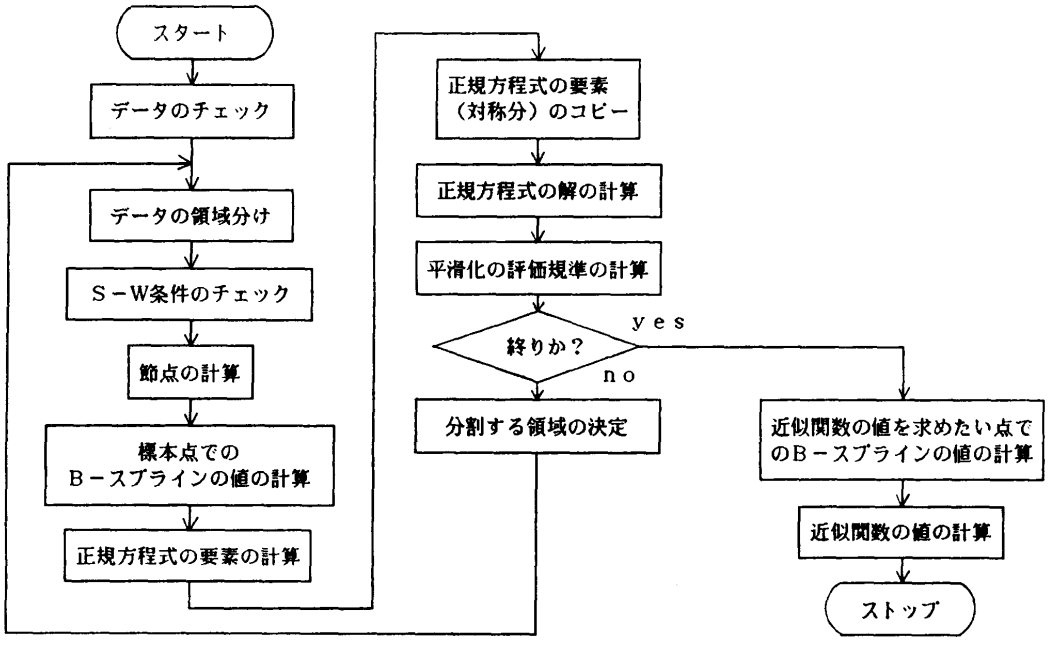

図 2 スプライン関数を用いた多次元ランダムデータ平滑化のアルゴリズム
Fig. 2 An algorithm of data fitting for multivariate random data with a spline function.

て，図1の計算手順を用いて同時に（並列に）必要なB-スプラインの値を計算できる．このようにすると，ベクトル長を標本点の数 N にすることができる．一般に，高速計算を必要とするような問題では N は大きいと考えてよいので，この方法はベクトル計算機に適合する．

なお，記憶容量を節約するため，図1において計算の順序を左から右へ，下から上へ，と行うことにすれば，一つの標本点 $x^{(q)}$ に対して大きさ m_q の配列を用意するだけでよい．なぜならば，図1の $M_{r,j}(x^{(q)})$ $(r=1,2,\cdots,m-1)$ および $N_{m,j}(x^{(q)})$ は，重ね書きできるからである．ゆえに，標本点の数が N であれば，すべてのB-スプラインの値を計算して格納するために必要な記憶容量は $N\times(m_1+m_2+\cdots+m_n)$ の配列だけとなる．

次に，正規方程式の要素の計算をベクトル計算機に適合させることを考えよう．これは，**各領域ごとにまとめて**正規方程式の要素を計算することによって実現できる．たとえば，2次元データの場合，各領域ごとに(17)で表される要素 $a_{u,v}$ の値を計算すればよい．このとき，ベクトル長は各領域内の標本点の数に等しくなる．高速計算を必要とする問題では，データの量は十分多いと考えてよいので，各領域内の標本点の数も多い．したがって，ベクトル長を十分長くできる．なお，複数個の領域にまたがる要素は関係する領域ごとの計算値の和になる．定数項の要素の計算も同様である．また，多次元の場合には上半分の要素の中にも対称な要素があるので，その点を考慮すると無駄な計算を省くことができる．

正規方程式の解の計算をベクトル計算機に適合させる点については，そのためにどのような解法を用いるかによって異なる．ここでは修正コレスキー法を適用する場合について述べる．S-W 条件が成り立つときには，正規方程式の係数行列 A は正定値対称でランク落ちしない．したがって，たとえば A を $U^{\mathrm{T}}DU$ 分解して，

$$U^{\mathrm{T}}D\boldsymbol{y}=\boldsymbol{d} \tag{23}$$

$$U\boldsymbol{c}=\boldsymbol{y} \tag{24}$$

を順に解けばよい．A の $U^{\mathrm{T}}DU$ 分解は，A が対称かつ帯行列であることを利用して，上半分のみを長方形領域にストアして行うことができる．このとき，配列参照が連続アクセスになるように配慮する[7]．

平滑化の評価規準および近似関数値の計算の部分も，すべての標本点を用いて同時に（並列に）計算することができ，ベクトル計算機によく適合する．以上述べた部分のほかで演算量の多いところは，データの領域分けである．これは，正規方程式の要素の計算をベクトル計算機に適合させるために必要なものである．データの分類方法としては，クイックソートなどの算法がよく知られているが，ここではそれを適用しない．その理由の一つは，分点の作る小領域ごとにデータを分類するだけでよいからである．もう一つの理由は，クイックソートなどを使ったときの処理時間が相当大きいためである．ここでは，小領域の数だけスロットを用意しておき，各データをつぎつぎと該当するスロットへ入れていく方法を用いる．この方法は余分な記憶容量を必要とするが，処理時間はクイックソートなどに比べるとわずかである．

7. 計算結果

では，二，三の数値例を示すことにしたい．ここで用いた計算機は富士通のベクトル計算機（スーパコンピュータ）FACOM VP-200 である．計算は，すべて倍精度で行った*．

表1は，2次元の領域 $R=[0,1]\times[0,1]$ の中のランダムな点で与えた 10,000 個のデータに対して，5次のスプライン関数を用いて平滑化を行った場合の例である．この表の各項目は，データ平滑化のアルゴリズムの各部分を表している（図2参照）．また，表の上段はスカラ実行の時間を，下段はベクトル実行の時間を表している．スカラ実行の時間を見ると，B-スプラインの値の計算，正規方程式の要素の計算などで多くの時間を消費しているが，ベクトル実行の時間を見ると，これらの部分が大きく加速されていることが分かる．なお，スカラ実行の時間 (S) とベクトル実行の時間 (V) の比 (S/V) を加速率と呼ぶ．

表2は，3次元の領域 $R=[0,1]\times[0,1]\times[0,1]$ の中のランダムな点で与えた 64,000 個のデータに対して，3次のスプライン関数を用いてデータ平滑化を行った場合の例である．3次元になると，特に正規方程式の要素の計算のところで多くの時間を消費しているが，6章で述べたようにしてベクトル計算機にうまくのせることにより，大きく加速されていることが分かる．

以上の計算例からも分かるように，本論文で提案した方法を用いればベクトル計算の効果は大である．な

* 本論文の数値例の計算時間データは，1987年3月京都大学大型計算機センターにおいて測定したものである．

表 1　5次のスプライン関数を用いた2次元ランダムデータ平滑化の所要時間
Table 1　Timing results for fitting of two-dimensional random data with a spline function of fifth degree.

分点数（共通 x y 各軸）	各部分の処理時間（上段の数値はスカラ実行，下段の数値はベクトル実行，単位 msec）									合計	累積
	データのチェック									—	—
	12 16									12 16	12 16
	データの領域分け	S-W 条件のチェック	節点の計算	標本点での B-スプラインの計算	正規方程式の要素の計算	正規方程式の要素のコピー	正規方程式の解の計算	平滑化の評価規準の計算	分割する領域の決定	—	—
3	54	0	0	575	3341	2	7	382	0	4361	4373
	125	0	0	101	91	2	5	45	1	370	386
4	57	0	0	585	2637	2	14	382	0	3677	8050
	125	0	1	101	102	3	8	45	1	386	772
5	62	0	0	594	2360	3	24	382	1	3426	11476
	125	0	0	102	112	3	12	45	0	399	1171
6	65	0	0	603	2269	4	39	381	0	3361	14837
	125	1	0	102	126	5	17	45	0	421	1592
7	67	1	0	611	2267	4	59	382	0	3391	18228
	128	0	0	106	143	6	25	47	0	455	2047
8	71	0	0	620	2273	6	86	382	0	3438	21666
	130	1	0	105	163	7	34	46	1	487	2534
9	74	0	0	628	2282	7	120	382	0	3493	25159
	128	1	0	103	182	9	43	45	0	511	3045

表 2　3次のスプライン関数を用いた3次元ランダムデータ平滑化の所要時間
Table 2　Timing results for fitting of three-dimensional random data with a spline function of third degree.

分点数（軸共通 x y z 各）	各部分の処理時間（上段の数値はスカラ実行，下段の数値はベクトル実行，単位 msec）									合計	累積
	データのチェック									—	—
	113 146									113 146	113 146
	データの領域分け	S-W 条件のチェック	節点の計算	標本点での B-スプラインの計算	正規方程式の要素の計算	正規方程式の要素のコピー	正規方程式の解の計算	平滑化の評価規準の計算	分割する領域の決定	—	—
3	443	0	0	2528	61868	8	97	6464	0	71408	71521
	1141	0	0	435	1781	11	32	535	0	3935	4081
4	480	0	0	2626	59506	18	450	6467	0	69547	141068
	1142	0	0	439	1884	22	89	535	0	4111	8192
5	511	0	0	2718	43762	32	1468	6470	0	54961	196029
	1149	0	0	443	2080	39	214	535	0	4460	12652
6	541	0	0	2811	42456	53	3980	6474	0	56315	252344
	1148	0	0	447	2265	63	458	535	0	4916	17568
7	575	0	0	2906	41847	82	9481	6477	0	61368	313712
	1154	0	0	451	2575	97	901	535	0	5713	23281
8	603	0	0	2996	41794	121	20662	6480	0	72656	386368
	1165	0	0	455	3074	141	1734	535	0	7104	30385

お，ここで述べた計算例は，6章で述べたアルゴリズムの有効性を調べるため，ベクトル化制御行（最適化制御行）などによるチューニングを行っていないプログラムを用いた場合である．

8. むすび

本論文では，スプライン関数を用いた多次元ランダムデータの平滑化について，ベクトル計算機向きの算法を述べた．必要なB-スプラインの値は，各座標軸方向ごとに，あらかじめすべての標本点を与えて，集団並列的に計算できることが分かった．したがって，ベクトル計算機によく適合させられる．また，最も計算量の多い正規方程式の要素の計算は，あらかじめ各領域ごとに標本点を分類しておき，各領域ごとにまとめて計算することにより，ベクトル長を十分長くできることが分かった．その他の部分もベクトル計算機によく適合させることができ，高速な計算が可能となった．ここで述べた方法を用いれば，3次元データで，標本点の数が節点の数よりも十分多いとき，ベクトル化指示行によるチューニングを行わなくても 10 倍以上の加速率を達成できる（富士通のスーパコンピュータ VP-200 の場合）．

なお，ここで提案した方法は，標本点の数が節点の数に比べて十分多い場合に有効であることに注意しておきたい．また，格子点データの場合には，それ専用の算法を用いる方がよい（これについては，別に報告する予定である）．

今後の課題としては，正規方程式を立てないで，直交変換を用いて解く方法の場合のベクトル計算機との適合性を調べること，およびS-W条件が成り立たない場合の最小二乗解を求める方法のベクトル計算機版を研究開発することである．

謝辞 著者の一人吉本は，スーパコンピュータについて御教示いただいた京都大学工学部情報工学科国枝義敏氏に感謝します．本研究の一部は京都大学大型計算機センターの開発課題として行われた．

参考文献

1) de Boor, C.: *A Practical Guide to Splines*, p. 392, Springer-Verlag, New York (1978).
2) 市田，吉本：スプライン関数とその応用（シリーズ新しい応用の数学 20），p. 220, 教育出版，東京 (1979).
3) 津田：スーパーコンピュータの現状と将来，機械の研究，Vol. 38, pp. 559-564 (1986).
4) 吉本，津田：スプライン関数を用いた多次元データの平滑化—ベクトル計算機向きの算法—，第3回ベクトル計算機応用シンポジウム論文集，京都大学大型計算機センター，pp. 27-36 (1987).
5) 吉本，津田：ベクトル計算機に適したB-スプラインの計算法，京都大学数理解析研究所講究録第613号［スーパーコンピュータのための数値計算アルゴリズムの研究］，pp. 188-203 (1987).
6) 吉本，津田：ベクトル計算機に向いたB-スプラインの計算法とその応用，第34回情報処理学会全国大会論文集，pp. 45-46 (1987).
7) 森：FORTRAN 77 数値計算プログラミング，p. 342, 岩波書店，東京 (1986).

（昭和62年7月7日受付）
（昭和62年11月11日採録）

吉本富士市（正会員）

昭和18年生．昭和41年岡山大学工学部電気工学科卒業．同年明石工業高等専門学校助手．講師，助教授を経て昭和62年教授，現在に至る．専門共通科目（情報処理）担当．昭和52年工学博士（京都大学）．昭和59年11月～60年8月文部省在外研究員として米国パデュー大学などに出張．現在の研究テーマは，関数近似の問題（特に多変数問題）における並列数値計算とその応用など．

津田　孝夫（正会員）

1932年生．1957年京都大学工学部電気工学科卒業．現職は京都大学工学部情報工学科教授．工学博士．現在の主要研究テーマは，メモリ階層間データ転送量の下界とそれによるアルゴリズムの最適化，ベクトル計算機のための自動ベクトル化の自動並列化，実時間オペレーティングシステムなど専用OSの構成と実現法など．