

# 区分的多項式とスプライン関数の基礎

やさしい  
解説

——折れ線グラフを曲線にしてみよう——\*

北原大地 (大阪大学)\*\*

## 1. はじめに

スプライン関数とはどのような関数だろうか？とても簡単に言うと、スプライン関数は「折れ線グラフを曲線化した関数」である。身近な応用例に、コンピュータで文字を表示する際のフォントのデータがあり、なんと、各文字のすべての輪郭がスプライン関数を用いて作られているのである。本稿では、小学校で教わる折れ線グラフを起点として、それを曲線化したスプライン関数を、数式も交えつつ、なるべく分かり易く解説する。

### 2. 1次元データ補間

図-1のように横軸を  $x$  軸、縦軸を  $y$  軸として、黒丸で示した5個のデータ点  $(x_0, y_0) = (-3, 6)$ ,  $(x_1, y_1) = (-1, 2)$ ,  $(x_2, y_2) = (0, 1)$ ,  $(x_3, y_3) = (1, 4)$ ,  $(x_4, y_4) = (4, 7)$  が与えられている状況を考える。このとき、変数  $x$  と  $y$  の値がどちらも連続的に動くのならば、データ点の間の  $x$  と  $y$  の関係、例えば  $x = (x_2 + x_3)/2 = 0.5$  における  $y$  の値を知りたくなる場合がある。この際に利用する一番簡単な方法として、小学生のときに「折れ線グラフ」を教わっただろう。

図-1の点線で示した折れ線グラフは、隣り合うデータ点を結ぶ線分を定規を使って引いていけば小学生でも簡単に描ける。折れ線グラフは各  $x$  における  $y$  の推定値を示しているので「 $x$  を引数とする関数」と見なせる。このように、データ点を通りつつ、データ点の間をつなぐ関数を作ることを「補間」又は「内挿」という。折れ線グラフは補間法の一例である。補間の高精度化はアップサンプリング・数値積分・データ解析などの高精度

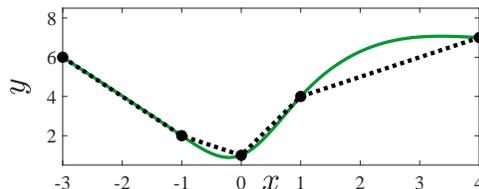


図-1 5個のデータ点(黒丸)とそれらを補間する折れ線グラフ(点線)及び3次スプライン関数(実線)

化に直結するため、等間隔でないデータ点にも対応できる高精度な補間法の確立が重要となる。

さて、図-1の折れ線グラフは、 $x_{i-1} \leq x \leq x_i$  ( $i = 1, 2, 3, 4$ )の範囲において、 $(x_{i-1}, y_{i-1})$  と  $(x_i, y_i)$  を結ぶ線分である。この線分を関数と見なして式で書くと、傾きが  $(y_i - y_{i-1}) / (x_i - x_{i-1})$  でデータ点  $(x_{i-1}, y_{i-1})$  を通る直線なので

$$y = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} (x - x_{i-1}) + y_{i-1} \quad (1)$$

と表せる。式(1)より、 $x = (x_2 + x_3)/2 = 0.5$  における  $y$  の値は、 $x_2 \leq x \leq x_3$  での線分から

$$\begin{aligned} y &= \frac{y_3 - y_2}{x_3 - x_2} (0.5 - x_2) + y_2 \\ &= \frac{4 - 1}{1 - 0} (0.5 - 0) + 1 = 2.5 \end{aligned}$$

のように簡単に推定できる。

図-1の折れ線グラフを見ると、各データ点の前後で線分の傾きが不連続に変化している。一方、世の中の多くの事象は折れ線グラフのようにカクカクと直線的には推移せずに、滑らかに曲線的に推移する。図-1の実線のようにデータ点を滑らかに結ぶ曲線を引くための道具として「自在定規」という定規が存在する(ちなみに筆者は使用した経験がない)。その一種に「スプライン」と呼ばれるものがあり<sup>1</sup>、これを使って引いた曲線を関数

\* Basics of piecewise polynomials and spline functions: Let's make line charts smooth.

\*\* Daichi Kitahara (Osaka University, Suita, 565-0871)  
e-mail: kitahara@eei.eng.osaka-u.ac.jp  
[doi:10.20697/jasj.78.10.1]

<sup>1</sup> コンピュータを用いた製図が普及した現在では、定規として実物のスプラインを使う機会は失われている。写真は <https://www.core77.com/posts/55368> で見られる。

として再現したものが「3次スプライン関数」である [1]。折れ線グラフやスプライン関数は「区分的多項式」であり、中学校や高校で教わる「多項式」へと分割することができる。以下では第3章で多項式の性質を確認し、その後第4章で区分的多項式を、第5章でスプライン関数を解説する。

### 3. 多項式

多項式の定義を確認する。まず0次多項式とは「すべての  $x$  に対して同じ値  $c_0$  を返す関数」

$$p(x) = c_0 \quad (2)$$

であり、定数関数ともいう。次に1次多項式は

$$p(x) = c_1x + c_0 \quad (3)$$

と表せる「直線」であり、傾きを意味する係数  $c_1$  と切片を示す係数  $c_0$  の二つのパラメータを持つ。例えば式 (1) では、 $c_1 = (y_i - y_{i-1}) / (x_i - x_{i-1})$ 、 $c_0 = y_{i-1} - x_{i-1}(y_i - y_{i-1}) / (x_i - x_{i-1})$  である。式 (3) は  $c_1 > 0$  ならば単調増加関数、 $c_1 < 0$  ならば単調減少関数であり、 $c_1 = 0$  のときは式 (2) の定数関数になる。直線に対して二つの独立な条件 (例：直線が通過する2点の  $xy$  座標) を課せば、二つの条件を両立する係数の組み合わせ  $(c_0, c_1)$  は一通りに限られるため、直線が一意に定まる。

0次多項式と1次多項式は直線しか表現できないが、 $x^2$  の項を加えて定義される2次多項式

$$p(x) = c_2x^2 + c_1x + c_0 \quad (4)$$

では「曲線」を表現できるようになる。式 (4) は  $c_2 > 0$  ならば下に凸の放物線に、 $c_2 < 0$  ならば上に凸の放物線になり、軸である  $x = -c_1/2c_2$  において最小値又は最大値をとる。また、 $c_2 = 0$  のときは式 (3) の1次多項式になる。2次多項式に対して三つの独立な条件 (例：曲線が通過する3点の  $xy$  座標) を課せば、三つの条件をすべて満足する係数の組み合わせ  $(c_0, c_1, c_2)$  は一通りに限られ、曲線 (放物線) の形状が一意に定まる。

上記を一般化すると、 $d$  次多項式とは  $d+1$  個の係数  $c_k$  ( $k = 0, 1, \dots, d$ ) をパラメータとして

$$p(x) = \sum_{k=1}^d c_k x^k + c_0 \quad (5)$$

と表せる関数である。 $d$  次多項式の厳密な定義は

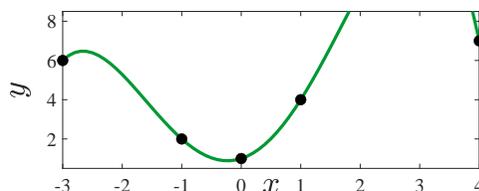


図-2 5個のデータ点を補間する唯一の4次多項式

条件に  $c_d \neq 0$  を必要とするが、本稿では  $c_d = 0$  の場合も式 (5) に含めることで「次数が  $d$  以下の多項式」を表すものとする。このような場合には式 (5) を「高々  $d$  次の多項式」と呼ぶ。式 (5) の多項式に対して  $d+1$  個の独立な条件を課せば、すべてを満足する係数の組み合わせ  $(c_0, c_1, \dots, c_d)$  は一通りに限られ、関数の形状が一意に定まる。

図-1の5個のデータ点を補間する多項式を考えてみる。任意に与えられた5個のデータ点を必ず通過できるように、次数  $d$  を4以上にする必要はある。特に  $d = 4$  のときは、補間条件  $f(x_i) = y_i$  ( $i = 0, 1, 2, 3, 4$ ) から得られる以下の連立方程式

$$x_i^4 c_4 + x_i^3 c_3 + x_i^2 c_2 + x_i c_1 + c_0 = y_i \quad (i = 0, 1, 2, 3, 4) \quad (6)$$

の解として、係数  $(c_0, c_1, c_2, c_3, c_4)$  の値が一意に定まる。式 (6) において、 $c_k x_i^k$  ではなく  $x_i^k c_k$  と書いたのは、 $x_i^k$  がデータ点の値から決まる定数であり、式 (6) が5個の係数  $c_k$  に関しての方程式になることを認識し易くするためである。式 (6) の連立方程式を解き、5個のデータ点を補間する唯一の4次多項式を図-2に示す。この4次多項式はデータ点をすべて通る曲線ではあるが、データ点の間、特に  $1 \leq x \leq 4$  で不自然に大きく振動し、枠の外に出てしまっている。この振動は「ルンゲ現象」といい、次数  $d$  が上がるほど大きな振動が発生し易くなることが知られている。

### 4. 区分的多項式

図-2のように、データ点全体を単一の多項式で補間すると、多くの場合でルンゲ現象が起こり、補間結果に不自然な振動が生じてしまう。一方、図-1の折れ線グラフはカクカクとはしているが、不自然な振動を起こしていない。折れ線グラフの数式をもう一度見てみよう。図-1の折れ線グラフ全体を関数  $f(x)$  とすると、式 (1) から

$$f(x) = \begin{cases} -2x & (-3 \leq x \leq -1) \\ -x + 1 & (-1 \leq x \leq 0) \\ 3x + 1 & (0 \leq x \leq 1) \\ x + 3 & (1 \leq x \leq 4) \end{cases} \quad (7)$$

である。式 (7) から、図-1 では 1 次多項式を 4 個接続してデータ点を補間していたことが分かる。

第 3 章で直線 (1 次多項式) を曲線 ( $d$  次多項式) に一般化した要領で、式 (7) の折れ線グラフを一般化してみる。つまり、高々  $d$  次の多項式<sup>2</sup>

$$p_i(x) = \sum_{k=0}^d c_k^{(i)} x^k + c_0^{(i)} \quad (8)$$

を複数個 (ここでは  $n + 2$  個) 接続した関数

$$f(x) = \begin{cases} p_0(x) & (-\infty < x \leq \xi_0) \\ p_1(x) & (\xi_0 \leq x \leq \xi_1) \\ \vdots & \\ p_n(x) & (\xi_{n-1} \leq x \leq \xi_n) \\ p_{n+1}(x) & (\xi_n \leq x < \infty) \end{cases} \quad (9)$$

でデータ点を補間することを考えてみる。式 (9) で表される関数  $f(x)$  が「 $d$  次の区分的多項式」である<sup>3</sup>。多項式が切り替わる場所  $\xi_i$  は「節点」と呼ばれ、 $\xi_0 < \xi_1 < \dots < \xi_n$  を満たす。

節点の数  $n + 1$  とデータ点の数は必ずしも一致している必要はなく、節点  $\xi_i$  とデータ点の  $x$  座標も一致している必要はないが、簡単のために本稿では節点とデータ点の位置をそろえる。具体的に図-1 のような 5 個のデータ点に対しては、 $n = 4$ 、 $\xi_0 = x_0$ 、 $\xi_1 = x_1$ 、 $\xi_2 = x_2$ 、 $\xi_3 = x_3$ 、 $\xi_4 = x_4$  となる。式 (9) の外側の 2 区間  $(-\infty, x_0]$  と  $[x_n, \infty)$  ではデータ点の外の  $x$  と  $y$  の関係を推定する「外挿」が行われることになるが、何等かの強い根拠がない限りは外挿結果を当てにしてはならない。したがって、これ以降はデータ点の内側の  $n$  区間  $[x_{i-1}, x_i]$  ( $i = 1, 2, \dots, n$ ) の  $p_i(x)$  のみを扱う。各  $p_i(x)$  は  $d + 1$  個の係数  $c_k^{(i)}$  ( $k = 0, 1, \dots, d$ ) を持つので、区分的多項式の形状を決めるための係数 (パラメータ) の総数は  $(d + 1)n$  個である。

<sup>2</sup>式 (8) で上添え字の  $(i)$  は「 $p_i(x)$  の係数」を意味する。  
<sup>3</sup>節点  $x = \xi_i$  で値が不連続な関数 (例: 階段関数) を区分的多項式に含める場合もあるが、本稿では区分的多項式が全区間  $(-\infty, \infty)$  に渡って連続となるように、式 (9) で  $f(\xi_i) = p_i(\xi_i) = p_{i+1}(\xi_i)$  が成り立つとする。

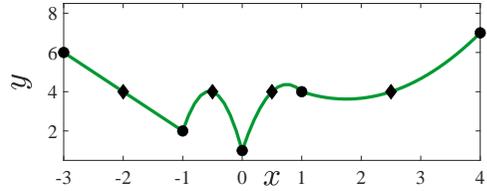


図-3 各区間の中央において  $y_i$  の平均値である 4 (ひし形で示した点) を通る 2 次の区分的多項式による補間

図-1 の 5 個のデータ点を補間する 1 次の区分的多項式を考えてみる。第  $i$  番目 ( $i = 1, 2, 3, 4$ ) の多項式  $p_i(x)$  のパラメータは  $(c_0^{(i)}, c_1^{(i)})$  であるが、二つの補間条件  $p_i(x_{i-1}) = y_{i-1}$  と  $p_i(x_i) = y_i$  を両立する値は  $c_1^{(i)} = (y_i - y_{i-1}) / (x_i - x_{i-1})$  かつ  $c_0^{(i)} = y_{i-1} - x_{i-1}(y_i - y_{i-1}) / (x_i - x_{i-1})$  のみである。この  $p_i(x)$  は式 (1) の線分そのものである。つまり、式 (7) の折れ線グラフは図-1 のデータ点を補間する唯一の 1 次の区分的多項式である。

さて、本来の目的に立ち返り、5 個のデータ点を曲線で補間することを考えよう。曲線を表すための最小次数は  $d = 2$  である。このとき  $f(x)$  は

$$f(x) = \begin{cases} c_2^{(1)} x^2 + c_1^{(1)} x + c_0^{(1)} & (x_0 \leq x \leq x_1) \\ c_2^{(2)} x^2 + c_1^{(2)} x + c_0^{(2)} & (x_1 \leq x \leq x_2) \\ c_2^{(3)} x^2 + c_1^{(3)} x + c_0^{(3)} & (x_2 \leq x \leq x_3) \\ c_2^{(4)} x^2 + c_1^{(4)} x + c_0^{(4)} & (x_3 \leq x \leq x_4) \end{cases} \quad (10)$$

となり、各  $p_i(x)$  の係数の数は 3 個である。一方、 $p_i(x)$  に対する条件は引き続き  $p_i(x_{i-1}) = y_{i-1}$  と  $p_i(x_i) = y_i$  の二つであり、この補間条件は

$$\begin{cases} x_{i-1}^2 c_2^{(i)} + x_{i-1} c_1^{(i)} + c_0^{(i)} = y_{i-1} \\ x_i^2 c_2^{(i)} + x_i c_1^{(i)} + c_0^{(i)} = y_i \end{cases} \quad (i = 1, 2, 3, 4) \quad (11)$$

と表せる。式 (11) では未知数に対して方程式の数が少ないので、係数の値が一意に定まらない。

係数を定めるには、条件を増やせばよい。試みに、 $p_i(x)$  がデータ点の中間で  $y_i$  の平均値になるという条件  $p_i((x_{i-1} + x_i) / 2) = (\sum_{i=0}^4 y_i) / 5 = 4$  を加えてみると、どうなるだろうか。この条件は

$$\left(\frac{x_{i-1} + x_i}{2}\right)^2 c_2^{(i)} + \frac{x_{i-1} + x_i}{2} c_1^{(i)} + c_0^{(i)} = 4 \quad (i = 1, 2, 3, 4) \quad (12)$$

と表せる。このように、 $(d-1)n$  個の追加条件を区分的多項式に与えれば、条件の数がパラメータの総数と同じ  $(d+1)n$  個となり、係数の値を決定できる。式 (11) と式 (12) から成る連立方程式を満たす唯一の 2 次の区分的多項式を図-3 に示す。この区分的多項式はデータ点と  $((x_{i-1} + x_i)/2, 4)$  を通る曲線であり、その振動は図-2 のルンゲ現象よりも小さい。しかし、1 本の曲線として自然な形には見えない。これは、図-1 の折れ線グラフと同じく、図-3 の区分的多項式  $f(x)$  の傾き  $f'(x)$  が各データ点の前後で不連続だからである。

## 5. スプライン関数

スプライン関数は、式 (9) の区分的多項式に微分に関する連続性条件を加えたものである。以下では、最も厳格な意味でのスプライン関数を紹介する [1-3]。式 (10) の 2 次の区分的多項式  $f(x)$  を滑らかな曲線にするためには、各節点  $x = x_i$  の前後で関数の傾き  $f'(x)$  を一致させる必要がある。区分的多項式は各区間で多項式なので、微分が

$$f'(x) = \begin{cases} 2c_2^{(1)}x + c_1^{(1)} & (x_0 \leq x \leq x_1) \\ 2c_2^{(2)}x + c_1^{(2)} & (x_1 \leq x \leq x_2) \\ 2c_2^{(3)}x + c_1^{(3)} & (x_2 \leq x \leq x_3) \\ 2c_2^{(4)}x + c_1^{(4)} & (x_3 \leq x \leq x_4) \end{cases} \quad (13)$$

のように簡単に計算できる。式 (13) が節点で連続になるには、 $f'(x_i) = p'_i(x_i) = p'_{i+1}(x_i)$ 、つまり

$$2x_i c_2^{(i)} + c_1^{(i)} = 2x_i c_2^{(i+1)} + c_1^{(i+1)} \quad (i = 1, 2, 3) \quad (14)$$

が成立すればよい。ここで、式 (14) の条件は両端の節点  $x_0$  と  $x_4$  を含まないことに注意されたい。

上記のように 2 次の区分的多項式の関数値  $f(x)$  と 1 階微分  $f'(x)$  がすべての節点で連続であるとき、 $f(x)$  を「2 次スプライン関数」と呼ぶ。同様に、 $d$  次の区分的多項式の関数値  $f(x)$  と 1 階微分から  $d-1$  階微分  $f^{(r)}(x)$  ( $r = 1, 2, \dots, d-1$ ) がすべて連続なら、 $f(x)$  を「 $d$  次スプライン関数」と呼ぶ。ここで、 $f^{(r)}(x)$  は  $r$  階微分を表し、 $f^{(1)}(x)$  は  $f'(x)$  と、 $f^{(2)}(x)$  は  $f''(x)$  と同じである。

図-1 のデータ点を補間する 2 次スプライン関数の条件は、式 (11) の  $2 \times 4 = 8$  個と、式 (14) の 3 個の計 11 個である。一方、式 (10) には係数  $c_k^{(i)}$

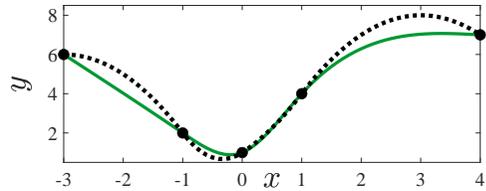


図-4 端条件  $f'(x_0) = 0$  の 2 次スプライン関数 (点線) と  $f''(x_0) = f''(x_4) = 0$  の 3 次スプライン関数 (実線)

が全部で 12 個あるので、係数の値を決めるためには何等かの条件をあと 1 個追加する必要がある。このような場合に、両端の節点  $x_0$  と  $x_n$  でのスプライン関数の 1 階微分や 2 階微分の値を指定することが多く [3]、これを「端条件」と呼ぶ。例えば、

$$f'(x_0) = 0 \quad (15)$$

という端条件を式 (10) の 2 次スプライン関数に課してみる。この条件は、式 (13) から

$$2x_0 c_2^{(1)} + c_1^{(1)} = 0 \quad (16)$$

と表される。式 (11) と式 (14) と式 (16) から成る連立方程式の解である 2 次スプライン関数を図-4 の点線に示す。この補間曲線はデータ点の前後で傾きが連続で振動も少ない。ただ、始点  $x = x_0$  での傾きを 0 に強制することは不自然と言える。

### 5.1 3 次自然スプライン関数とその最適性

2 次スプライン関数では端条件が一つとなり<sup>4</sup>、始点  $x_0$  か終点  $x_n$  のどちらか片方だけ傾きを指定するため曲線全体のバランスが悪くなり易い。そこで次数  $d$  を増やして、3 次スプライン関数による補間を試みる。各  $p_i(x)$  は 4 個の係数  $c_k^{(i)}$  を持ち、補間条件  $p_i(x_{i-1}) = y_{i-1}$  と  $p_i(x_i) = y_i$  は

$$\begin{cases} x_{i-1}^3 c_3^{(i)} + x_{i-1}^2 c_2^{(i)} + x_{i-1} c_1^{(i)} + c_0^{(i)} = y_{i-1} \\ x_i^3 c_3^{(i)} + x_i^2 c_2^{(i)} + x_i c_1^{(i)} + c_0^{(i)} = y_i \end{cases} \quad (i = 1, 2, 3, 4) \quad (17)$$

と表される。3 次スプライン関数の定義から、1 階微分  $f'(x)$  と 2 階微分  $f''(x)$  の両方が節点で連続でなければならない。この連続性条件はそれぞれ

$$\begin{aligned} 3x_i^2 c_3^{(i)} + 2x_i c_2^{(i)} + c_1^{(i)} &= 3x_i^2 c_3^{(i+1)} \\ &+ 2x_i c_2^{(i+1)} + c_1^{(i+1)} \quad (i = 1, 2, 3) \end{aligned} \quad (18)$$

<sup>4</sup>節点の中間にデータ点を置けば端条件を二つにできる。

と

$$3x_i c_3^{(i)} + c_2^{(i)} = 3x_i c_3^{(i+1)} + c_2^{(i+1)} \quad (i = 1, 2, 3) \quad (19)$$

のように表せる。3次スプライン関数全体で係数  $c_k^{(i)}$  は  $4 \times 4 = 16$  個存在するのに対して、条件は式(17)の8個、式(18)の3個、式(19)の3個の計14個しかないで、端条件が2個必要となる。

式(15)のような1階微分の条件は両端の形状に与える影響が大きいので、以下の2階微分の条件

$$f''(x_0) = 0 \quad \text{かつ} \quad f''(x_n) = 0 \quad (20)$$

を端条件とすることが多い。この2個の端条件は

$$3x_0 c_3^{(1)} + c_2^{(1)} = 0 \quad \text{かつ} \quad 3x_4 c_3^{(4)} + c_2^{(4)} = 0 \quad (21)$$

と表せる。3次スプライン関数  $f(x)$  が式(20)を満たすとき、 $f(x)$  を「3次自然スプライン関数」という。式(17)と式(18)と式(19)と式(21)から成る連立方程式の解である3次自然スプライン関数を図-4の実線に示す。これは図-1の実線と同じ曲線であり、2次スプライン関数よりも曲線の振動が小さく、両端での傾きも自然に見える。

3次自然スプライン関数によりデータ点を通るきれいな曲線が得られることが分かったが、これはあくまでも視覚的に良好であることを確認したに過ぎず、4次以上の区分的多項式など、他にも良い補間曲線があるかもしれない。そこで、補間曲線となる関数は「関数値  $f(x)$  と1階微分  $f'(x)$  及び2階微分  $f''(x)$  が連続であればよい」というように条件を変えてみる。この条件を満たすすべての関数の集合は  $C^2(\mathbb{R})$  と記される。集合  $C^2(\mathbb{R})$  は区分的多項式以外の関数も含んでいるが、驚くべきことに、以下のエネルギー尺度の意味で3次自然スプライン関数が最良であることを示せる。

**定理 1**  $n+1$  個のデータ点  $\{(x_i, y_i)\}_{i=0}^n$  が与えられているとする。ただし、 $x_0 < x_1 < \dots < x_n$  かつ  $n \geq 1$  である。このとき、以下の最小化問題

$$\begin{aligned} & \underset{f \in C^2(\mathbb{R})}{\text{minimize}} \int_{-\infty}^{\infty} |f''(x)|^2 dx \\ & \text{subject to} \quad f(x_i) = y_i \quad (i = 0, 1, \dots, n) \end{aligned} \quad (22)$$

の唯一の解は、節点が  $\xi_i = x_i$  ( $i = 0, 1, \dots, n$ ) の3次自然スプライン関数である [1, 2]。

式(22)は、データ点をすべて通るという条件の下で  $\int_{-\infty}^{\infty} |f''(x)|^2 dx$  の値を最小にする関数  $f(x)$  を集合  $C^2(\mathbb{R})$  の中から見つける問題である。2階微分の大きさ  $|f''(x)|$  は関数の傾き  $f'(x)$  の変化の大きさを表す。図-2のルンゲ現象のように関数が激しく振動している領域では、短い区間で関数の傾きが大きく変わることから、2階微分の二乗  $|f''(x)|^2$  の値が非常に大きくなる。一方で、曲線の形状が直線に近い領域では、傾きがほぼ一定値であるので、2階微分の二乗  $|f''(x)|^2$  が0に近くなる。このことから、 $\int_{-\infty}^{\infty} |f''(x)|^2 dx$  は「関数の局所振動エネルギー」と見なせる。これを補間条件の下で最小化する関数は、折れ線グラフを曲線化したような形状となり、それが正に3次自然スプライン関数なのである。定理1の証明は実はそれほど難しくない(例えば [1, 2, 4] を参照<sup>5)</sup>)。

### 5.2 3次スプライン関数の近似誤差

定理1の最適性の他にも、滑らかな関数を3次スプライン関数による補間で近似した際の誤差の上限が、以下のように具体的に解明されている。

**定理 2**  $g(x)$  は関数値と1階微分から4階微分までがすべて連続な関数で、 $n+1$  個のデータ点が  $\{(x_i, g(x_i))\}_{i=0}^n$  のように与えられているとする。ただし、 $x_0 < x_1 < \dots < x_n$  かつ  $n \geq 1$  である。 $f(x)$  はデータ点を補間する3次スプライン関数で、その節点は  $\xi_i = x_i$  ( $i = 0, 1, \dots, n$ ) であるとする。また、 $\bar{g}^{(4)} = \max_{x \in [x_0, x_n]} |g^{(4)}(x)|$ 、 $h_i = x_i - x_{i-1}$  ( $i = 1, 2, \dots, n$ )、 $\bar{h} = \max_i h_i$  とする。このとき、3次スプライン関数の端条件が  $f'(x_0) = g'(x_0)$  かつ  $f'(x_n) = g'(x_n)$ 、もしくは  $f''(x_0) = g''(x_0)$  かつ  $f''(x_n) = g''(x_n)$  ならば、

$$\max_{x \in [x_0, x_n]} |f(x) - g(x)| \leq \alpha \bar{g}^{(4)} \bar{h}^4$$

が成り立つ。ここで、 $\alpha = 5/384$  である [5]。

定理2は「データ点間隔  $h_i$  の最大値  $\bar{h}$  の四乗と同程度という速いスピードで誤差が小さくなる」ということを示しており、例えばデータ点間隔が半分になれば、誤差は  $(1/2)^4 = 1/16$  程度小さくなる。ただし、端条件を  $f''(x_0) = f''(x_n) = 0$  にして3次自然スプライン関数で補間を行う場合には、始点  $x = x_0$  と終点  $x = x_n$  付近では誤差がもう少し大きくなるので注意が必要である。

<sup>5)</sup>文献 [4] は <https://lab.d-kitahara.com> で読める。

## 6. スプライン関数の表現方法

スプライン関数の数式表現には、本稿で用いた式 (8) 及び式 (9) 以外にも、切断べき関数、差分商と B-スプライン、バーンスタイン・ベジエ形式など様々なものがあり、この表現方法の多様さが初学者に対する高い壁として立ちはだかる。表現を変更すると、議論や証明の分かり易さ、パラメータの意味と個数、計算アルゴリズムと計算時間、数値誤差の大きさなどが変わる。ただ、覚えておいてほしいのは、どの数式表現を用いても表現できるスプライン関数の形状は (ほぼ) 同じであり、結局は区分的多項式だということである。

以下では、スプライン関数の表現を変えて計算が簡単になる例を紹介する。式 (8) の上位互換として、 $x$  軸の原点を節点  $\xi_{i-1} = x_{i-1}$  に合わせて節点間隔  $h_i = \xi_i - \xi_{i-1} = x_i - x_{i-1}$  で割った<sup>6</sup>

$$p_i(x) = \sum_{k=1}^d \tilde{c}_k^{(i)} \left( \frac{x - x_{i-1}}{h_i} \right)^k + \tilde{c}_0^{(i)} \quad (23)$$

という表現がある。式 (23) では、 $x_{i-1} \leq x \leq x_i$  のときに  $\tau = (x - x_{i-1})/h_i$  の値が  $0 \leq \tau \leq 1$  を満たすため、べき乗の計算が常に 0 以上 1 以下の範囲となり、数値誤差を小さくできる。係数  $\tilde{c}_k^{(i)}$  の値を決める際は、例えば式 (17) の補間条件が

$$\tilde{c}_0^{(i)} = y_{i-1} \quad \text{かつ} \quad \tilde{c}_3^{(i)} + \tilde{c}_2^{(i)} + \tilde{c}_1^{(i)} + \tilde{c}_0^{(i)} = y_i \quad (i = 1, 2, 3, 4)$$

となり、式 (18) の 1 階微分の連続性条件が

$$h_{i+1}(3\tilde{c}_3^{(i)} + 2\tilde{c}_2^{(i)} + \tilde{c}_1^{(i)}) = h_i\tilde{c}_1^{(i+1)} \quad (i = 1, 2, 3)$$

となる。このように条件式から  $x_i^k$  が消えるので、式 (23) の  $\tilde{c}_k^{(i)}$  に関する連立方程式の方が、式 (8) の  $c_k^{(i)}$  に関する連立方程式よりも簡単に解ける。

式 (23) は任意の次数  $d$  に対して使える数式表現である。一方、最もよく用いるのは 3 次スプライン関数であり、 $d = 3$  の場合のみで使える以下の数式が有名である。3 次スプライン関数  $f(x)$  のパラメータを多項式の係数ではなく、節点  $x = x_i$  での 2 階微分値  $u_i = f''(x_i)$  ( $i = 0, 1, \dots, n$ ) とする。各  $p_i(x)$  は  $p_i(x_{i-1}) = y_{i-1}$ ,  $p_i(x_i) = y_i$ ,  $p_i'(x_{i-1}) = u_{i-1}$ ,  $p_i'(x_i) = u_i$  を満たすため、

$$p_i(x) = \frac{u_i - u_{i-1}}{6h_i} (x - x_{i-1})^3 + \frac{u_{i-1}}{2} (x - x_{i-1})^2 + \left( \frac{y_i - y_{i-1}}{h_i} - \frac{h_i(2u_{i-1} + u_i)}{6} \right) (x - x_{i-1}) + y_{i-1} \quad (24)$$

と表せる。さらに 3 次スプライン関数の定義から  $p_i'(x_i) = p_{i+1}'(x_i)$  であり、 $u_i$  に関する連立方程式

$$\begin{bmatrix} 2 & a_0 & & & & \\ b_1 & 2 & a_1 & & & \\ & b_2 & 2 & a_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{n-1} & 2 & a_{n-1} \\ & & & & b_n & 2 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} \quad (25)$$

が得られる。 $a_i, b_i, v_i$  ( $i = 1, 2, \dots, n-1$ ) は

$$\begin{cases} a_i = \frac{h_{i+1}}{h_i + h_{i+1}}, & b_i = \frac{h_i}{h_i + h_{i+1}} \\ v_i = 6 \frac{(y_{i+1} - y_i)/h_{i+1} - (y_i - y_{i-1})/h_i}{h_i + h_{i+1}} \end{cases}$$

と定義される。 $a_0, v_0, b_n, v_n$  は端条件が

$$f'(x_0) = e_0 \quad \text{かつ} \quad f'(x_n) = e_n$$

のとき  $a_0 = 1$ ,  $v_0 = 6[(y_1 - y_0)/h_1 - e_0]/h_1$ ,  $b_n = 1$ ,  $v_n = 6[e_n - (y_n - y_{n-1})/h_n]/h_n$  となり、

$$f''(x_0) = e_0 \quad \text{かつ} \quad f''(x_n) = e_n$$

のとき  $a_0 = 0$ ,  $v_0 = 2e_0$ ,  $b_n = 0$ ,  $v_n = 2e_n$  となる。式 (8) の  $c_k^{(i)}$  や式 (23) の  $\tilde{c}_k^{(i)}$  に関する連立方程式では、未知数 (パラメータ) が  $(d+1)n$  個あった。一方、式 (25) の  $u_i$  に関する連立方程式では、未知数が  $n+1$  個に減っているため解が求まり易くなっている。さらに、式 (25) の左辺の行列は「三重対角行列」という特殊な構造の行列であり、連立方程式の解を非常に高速に計算することができる [2]。方程式の解となる  $u_i$  を式 (24) に代入すれば、最終的な補間曲線が得られる。

再度注意しておくが、データ点を補間する 3 次スプライン関数を作成するときに、 $c_k^{(i)}$  に関する方程式、 $\tilde{c}_k^{(i)}$  に関する方程式、 $u_i$  に関する方程式のどれを解いても、理論的には同じ形状の曲線が求まる。変わるのは計算アルゴリズムであり、それに付随して計算時間や数値誤差などが変わる。処理の内容 (今回はデータ点の補間) と計算上のテクニックを区別して考えることが大切である。

<sup>6</sup>式 (23) の表現に正式名称はないが、筆者は「区間正規化形式 (interval normalization form)」と呼んでいる。

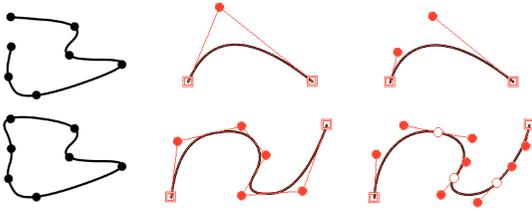


図-5 2次元平面上のデータ点を通る曲線及び閉曲線(左)と2次ベジェ曲線(中央)及び3次ベジェ曲線(右)

### 7. スプライン関数の応用技術

スプライン関数は1次元データの補間に使われるだけでなく、コンピュータ上での曲線や曲面の作成、データの平滑化や回帰分析などにも利用されている。これらの応用技術を簡単に紹介する。

#### 7.1 2次元平面上の曲線の作成

コンピュータ上で図-5のような曲線を描いた経験はないだろうか。実はこれらはスプライン関数を用いて作られている。図-5でも横を  $x$  軸、縦を  $y$  軸とする。図-5左上の曲線は、図-4の曲線とは異なり、同じ  $x$  座標に対して複数の  $y$  座標の値を示す箇所が存在するため、 $y = f(x)$  という形式では表現できない。そこで、媒介変数  $t$  を導入して曲線の  $x$  座標と  $y$  座標の動きをそれぞれ  $X(t)$  と  $Y(t)$  という別々のスプライン関数で表現する。

二つのスプライン関数  $X(t)$  と  $Y(t)$  は  $n+1$  個の節点  $\xi_i = i$  ( $i = 0, 1, \dots, n$ ) を共有しており、曲線の始点と終点は  $(X(0), Y(0)) = (x_0, y_0)$  と  $(X(n), Y(n)) = (x_n, y_n)$  とする。曲線の節点と補間するデータ点  $\{(x_i, y_i)\}_{i=0}^n$  の位置をそろえれば、 $X(i) = x_i$  かつ  $Y(i) = y_i$  ( $i = 0, 1, \dots, n$ ) という補間条件が得られ、これに適当な端条件を与えれば  $X(t)$  と  $Y(t)$  の形状が一意に定まる。

一方、始点と終点が  $(x_0, y_0) = (x_n, y_n)$  のように重なる「閉曲線」の場合は注意が必要となる。図-5左下の閉曲線のような自然な形になるには、 $X(0) = x_0 = x_n = X(n)$  と  $Y(0) = y_0 = y_n = Y(n)$  だけでは不十分で、 $X^{(r)}(0) = X^{(r)}(n)$  と  $Y^{(r)}(0) = Y^{(r)}(n)$  ( $r = 1, 2, \dots, d-1$ ) も満たす必要がある。このように、始点と終点においても関数値と1階微分から  $d-1$  階微分までが連続な  $d$  次スプライン関数を「周期スプライン関数」という。周期スプライン関数  $X(t)$  と  $Y(t)$  は端条件を必要とせず、補間条件のみで形状が定まる [1]。

実用上は、補間するデータ点  $\{(x_i, y_i)\}_{i=0}^n$  の値が決まっておらず、手作業で少しずつ所望の曲線を描きたい場合もある。この際には、 $n+2$  個の「制御点」 $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=0}^{n+1}$  と呼ばれる点と「2次のB-スプライン」 $B_i(t)$  ( $i = 0, 1, \dots, n+1$ ) という2次スプライン関数を用いて、 $X(t)$  と  $Y(t)$  を

$$X(t) = \sum_{i=0}^{n+1} \tilde{x}_i B_i(t) \quad \text{かつ} \quad Y(t) = \sum_{i=0}^{n+1} \tilde{y}_i B_i(t) \quad (0 \leq t \leq n) \quad (26)$$

とすることが多い(図-5中央)。式(26)は  $n=1$  のときに以下の「バーンスタイン・ベジェ形式」

$$X(t) = \tilde{x}_0(1-t)^2 + 2\tilde{x}_1t(1-t) + \tilde{x}_2t^2$$

と呼ばれる式になる。 $n \geq 2$  のときは各  $B_i(t)$  が

$$B_0(t) = (1-t)^2 \quad (0 \leq t \leq 1)$$

$$B_1(t) = \begin{cases} -\frac{3}{2}(t-\frac{2}{3})^2 + \frac{2}{3} & (0 \leq t \leq 1) \\ \frac{1}{2}(2-t)^2 & (1 \leq t \leq 2) \end{cases}$$

$$B_i(t) = \begin{cases} \frac{1}{2}(t-i+2)^2 & (i-2 \leq t \leq i-1) \\ -(t-\frac{2i-1}{2})^2 + \frac{3}{4} & (i-1 \leq t \leq i) \\ \frac{1}{2}(i+1-t)^2 & (i \leq t \leq i+1) \end{cases} \quad (i = 2, 3, \dots, n-1)$$

$$B_n(t) = \begin{cases} \frac{1}{2}(t-n+2)^2 & (n-2 \leq t \leq n-1) \\ -\frac{3}{2}(t-n+\frac{2}{3})^2 + \frac{2}{3} & (n-1 \leq t \leq n) \end{cases}$$

$$B_{n+1}(t) = (t-n+1)^2 \quad (n-1 \leq t \leq n)$$

のように定義される。ただし、各  $B_i(t)$  の値は、 $t$  が条件の範囲外するとき  $B_i(t) = 0$  である。式(26)の曲線は2次ベジェ曲線やB-スプライン曲線と呼ばれ、始点と終点が  $(X(0), Y(0)) = (\tilde{x}_0, \tilde{y}_0)$  と  $(X(n), Y(n)) = (\tilde{x}_{n+1}, \tilde{y}_{n+1})$ 、他の節点では  $(X(i), Y(i)) = ((\tilde{x}_i + \tilde{x}_{i+1})/2, (\tilde{y}_i + \tilde{y}_{i+1})/2)$  のように制御点の midpoint を通る ( $i = 1, 2, \dots, n-1$ )。

次数が  $d=3$  のときは、B-スプラインを式(26)のように足す方法が手作業に向かなくなるため、制御点が4個のバーンスタイン・ベジェ形式

$$X(t) = \tilde{x}_0(1-t)^3 + 3\tilde{x}_1t(1-t)^2 + 3\tilde{x}_2t^2(1-t) + \tilde{x}_3t^3 \quad (0 \leq t \leq 1)$$

の曲線をつなげて3次スプライン関数や区分的多項式<sup>7</sup>を作成する方法が一般的である(図-5右)。

<sup>7</sup>通常は、高々1階微分までの連続性しか考慮されない。

冒頭で書いたように、フォントのデータにおける文字の輪郭も 2 次又は 3 次のスプライン関数を用いて作られている。文字の輪郭を関数で表現することで輪郭が幾らでも拡大可能になり、サイズが変わっても美しい文字を表示できるのである。

## 7.2 誤差を含んだデータの平滑化

ここまでは「各データ点  $(x_i, y_i)$  に誤差がない」ことを前提としていた。しかしながら、実際にはデータ点  $(x_i, y_i)$  が何等かの測定誤差  $\epsilon_i$  を含んでいる場合も多い。このとき、 $x_i$  と  $y_i$  と  $\epsilon_i$  の関係は、 $y_i = g(x_i) + \epsilon_i$  のように表される。ここで、 $g(x)$  は「誤差がないときの  $x$  と  $y$  の関係」を表す関数である。たとえ  $g(x)$  が振動の少ない滑らかな関数であったとしても、誤差を含んだデータ点を 3 次スプライン関数で補間すると、図-6 の点線で示したように補間曲線は激しく振動してしまう。

本来の滑らかな関数  $g(x)$  を高精度に再現するには、各データ点を通る曲線ではなく、各データ点の近くを通りつつも振動は少ない曲線を作ればよい。このように、誤差を含んだデータ点から  $x$  と  $y$  の滑らかな関係を復元することを「平滑化」という。定理 1 では「補間」を式 (22) の最小化問題として定式化した。同じ要領で「平滑化」は、

$$\text{minimize}_{f \in C^2(\mathbb{R})} \sum_{i=0}^n |y_i - f(x_i)|^2 + \lambda \int_{-\infty}^{\infty} |f''(x)|^2 dx$$

と定式化される。これは、 $\sum_{i=0}^n |y_i - f(x_i)|^2 + \lambda \int_{-\infty}^{\infty} |f''(x)|^2 dx$  を最小にする関数  $f(x)$  を集合  $C^2(\mathbb{R})$  から見つける問題である。実はこの問題の唯一の解も、節点が  $\xi_i = x_i$  ( $i = 0, 1, \dots, n$ ) の 3 次自然スプライン関数となる [4]。各データ点との二乗誤差  $|y_i - f(x_i)|^2$  と局所振動エネルギー  $\int_{-\infty}^{\infty} |f''(x)|^2 dx$  の両方を小さくするので、平滑化パラメータ  $\lambda > 0$  の値が適切ならば<sup>8</sup>、図-6 の実線で示したような滑らかな曲線が求まる。

筆者はこの「スプライン平滑化」を発展させ、確率密度関数やパーセントイル曲線、不連続点を含む関数などを推定している [6-9]。スプライン関数は区分的多項式なので、高い表現能力を持つだけでなく、微分や定積分も簡単に計算できる。これにより、様々な関数の推定問題を「係数  $c_k^{(i)}$  に関する最小化問題」として解けるのである<sup>9</sup>。

<sup>8</sup>  $\lambda$  を大きくすると平滑化結果は直線に近づき、 $\lambda$  を 0 に近づけると平滑化結果は定理 1 の解の補間曲線に近づく。

<sup>9</sup> 係数に関する最小化問題の導出は文献 [4] に書いてある。

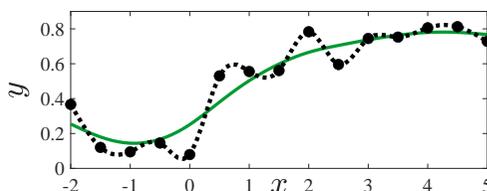


図-6 誤差を含んだデータに対する補間と平滑化の比較

## 8. おわりに

本稿では、折れ線グラフが 1 次の区分的多項式で、それを滑らかにした曲線がスプライン関数であることを説明した。さらに、スプライン関数の最適性、表現方法、応用も簡単に紹介した。本稿が、スプライン関数に興味を抱き、実際に使ってみる切っ掛けとなれば、うれしい限りである<sup>10</sup>。

## 文 献

- [1] 桜井 明, スプライン関数入門 (東京電機大学出版局, 東京, 1981).
- [2] 市田 浩三, 吉本 富士市, スプライン関数とその応用 (教育出版, 東京, 1979).
- [3] 菅野 敬祐, 吉村 和美, 高山 文雄, C によるスプライン関数: データ解析/CG/微分方程式 (東京電機大学出版局, 東京, 1993).
- [4] 北原 大地, “スプライン関数の基礎と分位点回帰への応用,” 信学技報, pp. 37-42 (2020).
- [5] C. A. Hall, “Optimal error bounds for cubic spline interpolation,” *J. Approx. Theory*, **16**, 105-122 (1976).
- [6] D. Kitahara and I. Yamada, “Probability density function estimation by positive quartic  $C^2$ -spline functions,” *Proc. IEEE ICASSP*, pp. 3556-3560 (2015).
- [7] D. Kitahara, L. Condat and A. Hirabayashi, “One-dimensional edge-preserving spline smoothing for estimation of piecewise smooth functions,” *Proc. IEEE ICASSP*, pp. 5611-5615 (2019).
- [8] D. Kitahara, K. Leng, Y. Tezuka and A. Hirabayashi, “Simultaneous spline quantile regression under shape constraints,” *Proc. EUSIPCO*, pp. 2423-2427 (2020).
- [9] S. Yao, D. Kitahara, H. Kuroda and A. Hirabayashi, “Modal interval regression based on spline quantile regression,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, to appear (2023).



北原 大地

東京工業大学情報工学科を 2012 年に卒業。同大学院集積システム専攻修士課程、通信情報工学専攻博士課程を 2014 年、2017 年に修了。2017 年 4 月より立命館大学情報理工学部助教、2022 年 4 月より大阪大学大学院工学研究科特任研究員。信号処理や気象レーダの研究に従事。

<sup>10</sup> 執筆の機会を提供して下さった矢田部浩平博士、並びに日本音響学会編集委員会の皆様に感謝申し上げます。