

スプライン補間のプログラム

山本昌志*

2005年1月18日

1 プログラム方法

理論から分かるように、スプライン補間を行うためには以下の2つの計算が必要である。

- 各点での2次導関数の値 u_i の計算
- 係数 a, b, c, d を求めて、補間の値の計算

それぞれ、C言語の関数を作成することにする。それぞれの関数の作成方法について、以下に述べる。

1.1 2次導関数の値を計算する関数

まず最初の関数で、標本点 x_i と y_i から、 x_i での2次導関数の値 u_i を計算するプログラムである。この関数は、最初に1回呼び出すだけでよく、結果は補間値を計算する次の関数に渡す。この2次導関数の値を計算するプログラムは、独立した関数としてモジュール化するのが常套手段である。そのほうが、プログラムが分かりやすくなり、メンテナンスも容易である。

C言語の関数でこれを実現するための関数のプロトタイプ宣言は、

```
void spline_cal_u(int n, double x[], double y[], double u[]);
```

のようにすればよい。nは標本点の最大の番号である。標本点の番号は、0から始まりnで終わる。配列x[]とy[]に標本のxとy座標の値を入れる。配列u[]は計算された2次導関数の値が入る。

この関数での処理の内容は、

- 標本点での2次導関数の値 u_i が満たす連立方程式を作成する。
- その連立方程式を解く。

*独立行政法人 秋田工業高等専門学校 電気工学科

である。要するに連立方程式を作り、そして解いているのである。解くべき連立方程式は、次の通りである。

$$\begin{pmatrix} 2(h_0 + h_1) & h_1 & & & & & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & & & & \\ & h_2 & 2(h_2 + h_3) & h_3 & & & \\ & & & \ddots & & & \\ & & & h_{j-1} & 2(h_{j-1} + h_j) & h_j & \\ & 0 & & & & \ddots & \\ & & & & & & h_{N-2} & 2(h_{N-2} + h_{N-1}) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_j \\ \vdots \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_j \\ \vdots \\ v_{N-1} \end{pmatrix} \quad (1)$$

ただし、 h_j と v_j は以下のとおり。

$$h_j = x_{j+1} - x_j \quad (j = 0, 1, 2, \dots, N-1) \quad (2)$$

$$v_j = 6 \left[\frac{y_{j+1} - y_j}{h_j} - \frac{y_j - y_{j-1}}{h_{j-1}} \right] \quad (j = 1, 2, \dots, N-1) \quad (3)$$

この関数のフローチャートを図1に示す。この関数では、連立方程式を解く必要がある。この連立方程式は、以下の特徴がある。

- 係数は、3重対角行列である。
- 対角成分は、ゼロにならない。

通常はLU分解、あるいは反復法で解くことになる。しかし、そのプログラムを書くのも面倒なので、以前作成したピボット選択のないガウス・ジョルダンを使えばよい。連立方程式により解かれた2次導関数の値は、配列 $u[]$ に入れられて呼び出し側に戻る。

連立方程式により計算される2次導関数の値は、 $u_1, u_2, u_3, \dots, u_{n-1}$ である。両端は自然境界条件で、 $u_0 = 0$ $u_n = 0$ とする。ここの関数では最後にその条件を配列を $u[0]=0$, $u[n]=0$ とすることで実現している。

1.2 補間の値を計算する関数

2次導関数の値が分かったので、残りの処理は、その値を利用して任意の x の補間値を求めることである。スプライン補間は、3次の区分多項式で標本点の間を補完するのは今まで述べたとおりである。そのた

めには、標本点の座標と2次導関数の値、即ち x_i, y_i, u_i がわかれば計算できる。2次導関数の値 u_i は予め、先の関数 `spline_cal_u(n, x, y, u)` で計算しておくものとする。

スプライン補間の計算に必要な x_i, y_i, u_i から、補間点 x の値を求める関数を作ろう。C言語の関数でこれを実現するための関数のプロトタイプ宣言は、

```
double spline(int n, double x[], double y[], double u[], double xx);
```

のようにすればよい。この関数の実引数 `xx` に補間値を求めたい x を入れるのである。そして、この関数の戻り値が x での補間値 y となる。

この関数の処理の内容は、

- 補間値を求める x が、どの標本点の間にあるか探す。
- 見つかった標本点の場所から、3次関数の係数を計算する。
- 3次関数から x の時の値、 y を計算する。

である。ここで重要なことは、 x がある区間を探し、3次関数の係数を求めることである。2次導関数と標本点の値から、それらの係数は

$$a = \frac{u_{j+1} - u_j}{6(x_{j+1} - x_j)} \quad (4)$$

$$b = \frac{u_j}{2} \quad (5)$$

$$c = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{1}{6}(x_{j+1} - x_j)(2u_j + u_{j+1}) \quad (6)$$

$$d = y_j \quad (7)$$

と計算できる。これらの係数から、補間の値 y は

$$y = ax^3 + bx^2 + cx + d \quad (8)$$

となる。

この関数のフローチャートを図2に示す。まず初めに、 x が存在する区間を2分探索により探している。これがわかれば、 x を挟む場所の x, y, u の値がわかる。そうすると、式(4)~(7)を用いて、3次関数の補間式の係数 a, b, c, d の値が計算できる。そして、式(8)から補間値 y がわかる。補間したい点 x が変わる毎に `spline(n, x, y, u, xx)` を呼び出せばよいわけである。

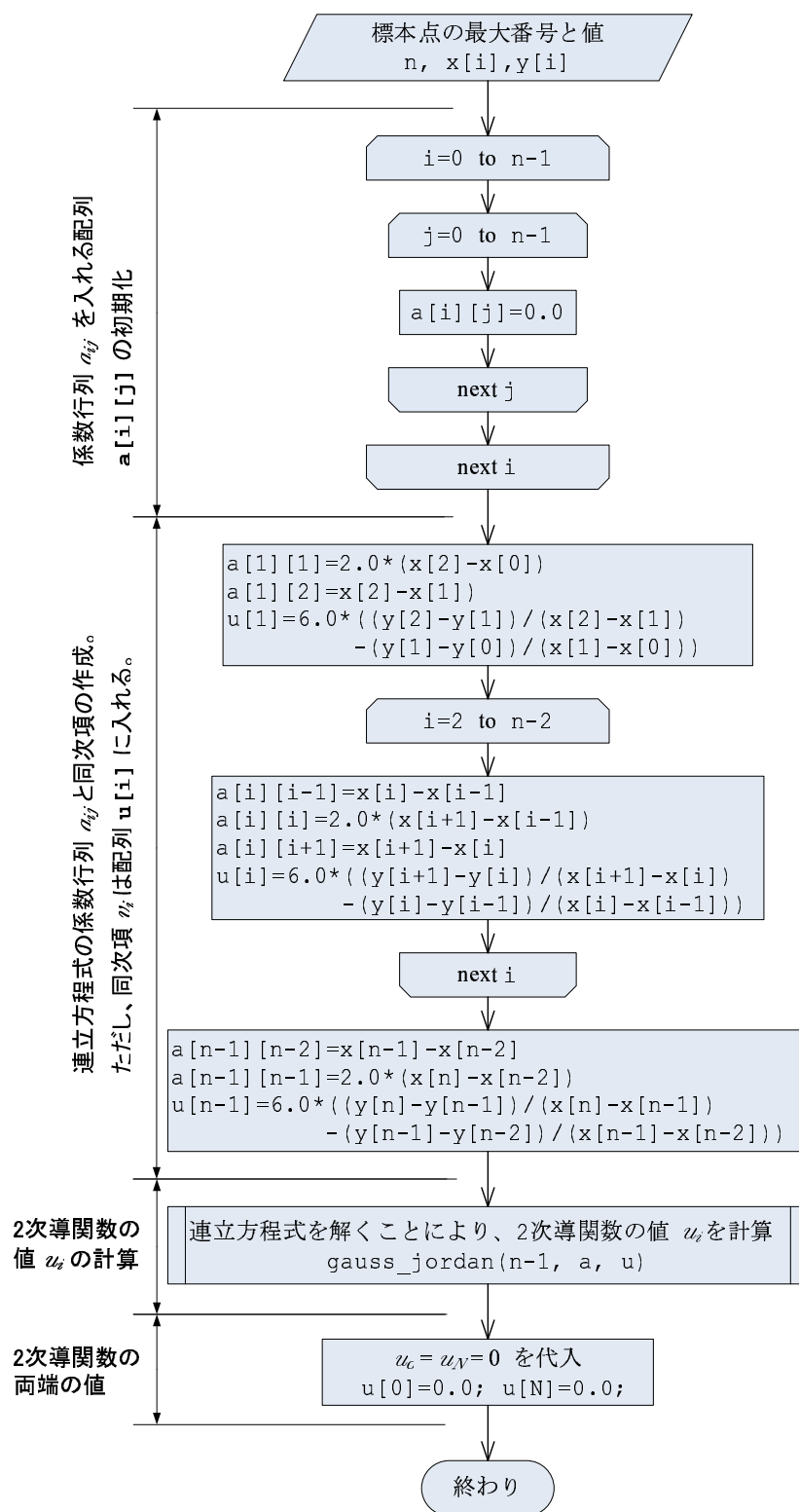


図 1: 2次導関数の u_i を求める関数。

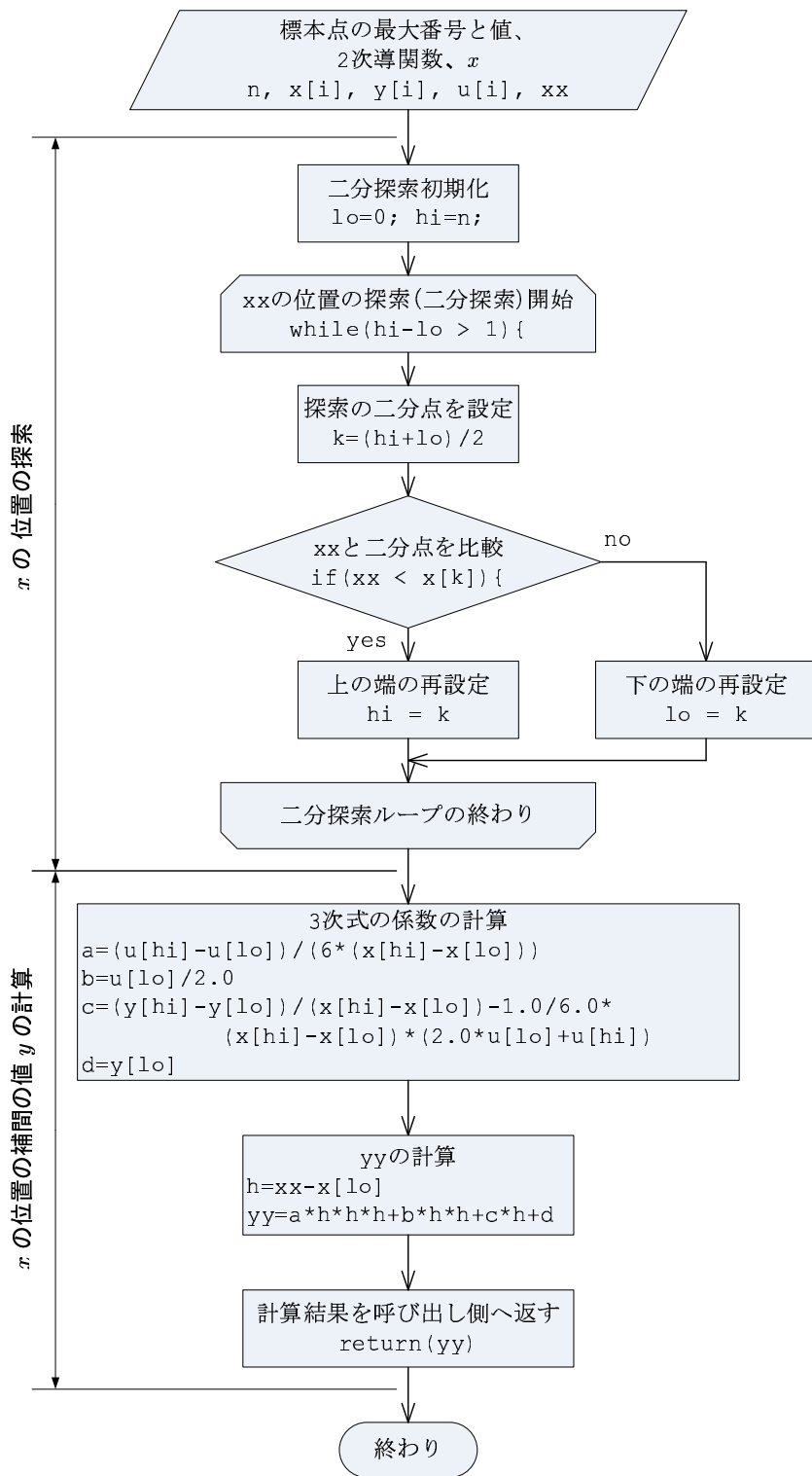


図 2: 補間された値を求める関数。