```
    simpson := 3*h/8*s
end;
```

【実行例】

$\int_0^\pi \sin x dx$ を，区間 $[0,\pi]$ をそれぞれ 9 等分，15 等分して求めたものについて示しておく（真値はもちろん 2 である）。

```
INTEGRAL FROM      0 TO    3.1416
UNDER     9 PARTITION=     2.0004E+00

INTEGRAL FROM      0 TO    3.1416
UNDER    15 PARTITION=     2.0000E+00
```

（注意） 本書の(29)の式はもう少し整理すれば

$$\frac{3h}{8}\left(f(x_0)+\sum_{i=0}^{m-1}(3f(x_{3i+1})+3f(x_{3i+2})+2f(x_{3i+3}))-f(x_{3m})\right)$$

となるから，これを用いれば $x_{3i}(i=1,\dots,m-1)$ の点で関数値を 2 度計算する手間が省ける。

3.3.2 スプライン関数を用いる方法†

ニュートン・コーツ系の公式では関数値は等間隔分点の上で与えられなければならないという条件がある。したがって被積分関数は一般に積分区間内の任意の点でその関数値が計算できる陽的な関数として与えられなければならない。また積分区間 $[a,b]$ を各小区間に分割しそれら各区間の上で多項式の定積分を求めていく場合，これら多項式が各小区間の境界の点で滑らかに接続されるという保証はどこにもない。すなわち各小区間の境界の点で隣り合う 2 つの多項式の傾きが一致している（言い換えれば導関数が連続）という保証はない(図3.9 参照)。

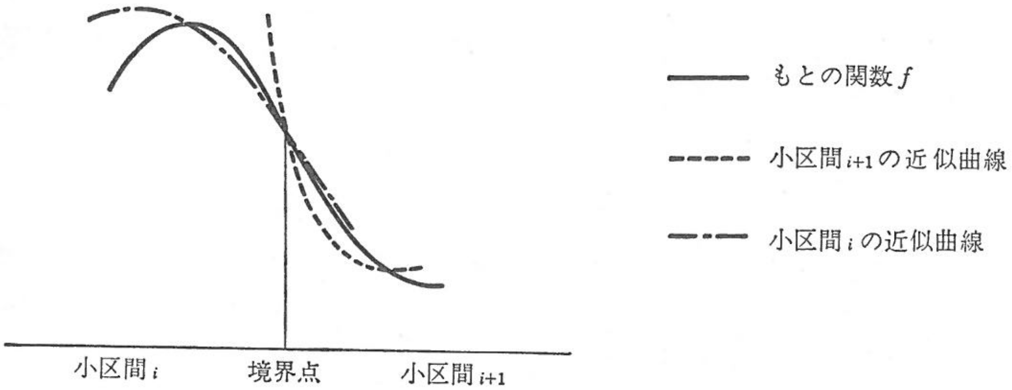

図 3.9 ニュートン・コーツ系の公式では 2 つの近似曲線は滑らかに接続されない

† この部分は筆者が「数学セミナー」(1974年 11月号)の NOTE 欄に書いたものをさらに詳細補足説明したものである。

スプライン関数を用いる方法とは，一般に不等間隔で観測されたデータの組(x_1, y_1)，(x_2, y_2), ..., (x_n, y_n)が与えられたとき，それらを通る滑らかな近似曲線を作り，それを積分するものである。正確にいうと次のようになる。

(i)　各小区間 $[x_i, x_{i+1}]$$(i=1, ..., n-1)$ を $(2k-1)$ 次の多項式f_iで近似する。

(ii)　各$f_i(i=1, 2, ..., n-1)$ は点 (x_i, y_i), (x_{i+1}, y_{i+1}) を通る。

(iii)　境界の点$x_i(i=2, ..., n-1)$ ではf_{i-1}とf_iの $(2k-2)$ 次までの導関数が連続。端点x_1, x_nにおいてはf_1とf_{n-1}のk次から $(2k-2)$ 次までの微分係数が0。

(iv)　$n \geq k \geq 1$。

このような$f_i(i=1, 2, ..., n-1)$ をつなぎ合わせて構成される関数$f(x)$を，点$(x_i, y_i)(i=1, 2, ..., n)$ を通る区間 $[x_1, x_n]$ 上の**スプライン関数**と呼ぶ(**図3.10**参照)。スプライン関数は我々が実際に手書きで描くような滑らかな曲線に対応しており，実際の目的のためには重要である。

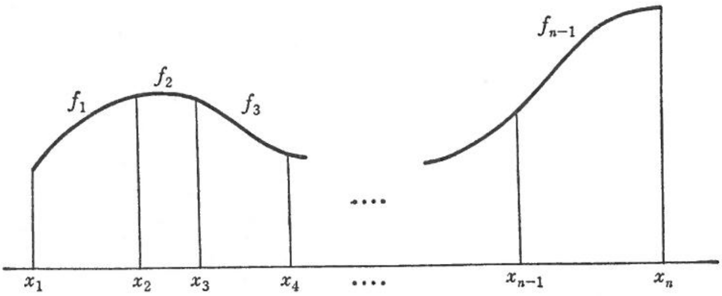

図3.10　スプライン関数

スプライン関数は普通$k=2$が使われる。すなわち各f_iを3次式で近似する。また端点x_1, x_nにおける微分係数は上の(iii)の後半の条件によればf_1とf_{n-1}の2次微分係数を0にすることになるが，普通これは1次微分係数（すなわち傾き）を外部の観測条件からそれぞれa, bと与える。以上の条件のもとでスプライン関数を構成しよう。

まず$f_i(i=1, 2, ..., n-1)$ を次のようにおく。

$$f_i(x)=y_i+A_i\left(\frac{x-x_i}{x_{i+1}-x_i}\right)+B_i\left(\frac{x-x_i}{x_{i+1}-x_i}\right)^2+C_i\left(\frac{x-x_i}{x_{i+1}-x_i}\right)^3 \tag{31}$$

これはx_iで値y_iをとるように構成されている3次式である。すなわち(ii)の点(x_i, y_i) を通るという条件が満たされている。問題はその他の条件から係数A_i, B_i, C_iを決めることである。(ii)の点(x_{i+1}, y_{i+1})を通るという条件は

$$f_i(x_{i+1})=y_{i+1}$$

であるから

$$y_i+A_i+B_i+C_i=y_{i+1} \quad (i=1, ..., n-1) \tag{32}$$

となる。

(31)の式を微分すると

$$f_i'(x)=\frac{A_i}{\Delta_i}+\frac{2B_i}{\Delta_i^2}(x-x_i)+\frac{3C_i}{\Delta_i^3}(x-x_i)^2 \quad (\text{ただし}\Delta_i=x_{i+1}-x_i) \tag{33}$$

となるから，したがって(iii)の条件の1階導関数が境界の点で連続ということは

$$f'_i(x_{i+1})=f'_{i+1}(x_{i+1})$$

より

$$\frac{A_i}{\Delta_i}+\frac{2B_i}{\Delta_i}+\frac{3C_i}{\Delta_i}=\frac{A_{i+1}}{\Delta_{i+1}} \quad (i=1, \dots, n-2) \tag{34}$$

となる。(33)をさらに微分すると

$$f_i''(x)=\frac{2B_i}{\Delta_i^2}+\frac{6C_i}{\Delta_i^3}(x-x_i) \tag{35}$$

したがって(iii)の2階導関数が境界の点で連続ということは

$$f_i''(x_{i+1})=f_{i+1}''(x_{i+1})$$

より

$$\frac{2B_i}{\Delta_i^2}+\frac{6C_i}{\Delta_i^2}=\frac{2B_{i+1}}{\Delta_{i+1}^2} \quad (i=1, \dots, n-2) \tag{36}$$

となる。さらに端点 x_1, x_n においては1次微分係数をそれぞれ a, b と与えるのであるから

$$f_1'(x_1)=a, \quad f_{n-1}'(x_n)=b$$

より

$$\frac{A_1}{\Delta_1}=a, \quad \frac{A_{n-1}}{\Delta_{n-1}}+\frac{2B_{n-1}}{\Delta_{n-1}}+\frac{3C_{n-1}}{\Delta_{n-1}}=b \tag{37}$$

となる。

(32), (34), (36), (37)に着目すると式の数は $(3n-3)$ 個ある。一方未知数 A_i, B_i, C_i $(i=1, \dots, n-1)$ も $3(n-1)$ 個であるから，以上の条件より A_i, B_i, C_i は唯1通りに定まる。式をもう少し整理すると，(34)の A_i に(32)の A_i を代入して

$$B_i+2C_i-\frac{\Delta_i}{\Delta_{i+1}}A_{i+1}=-(y_{i+1}-y_i) \quad (i=1, \dots, n-2) \tag{38}$$

また(36)の B_i に(38)の B_i を代入して

$$C_i+\frac{\Delta_i}{\Delta_{i+1}}A_{i+1}-\left(\frac{\Delta_i}{\Delta_{i+1}}\right)^2 B_{i+1}=y_{i+1}-y_i \quad (i=1, \dots, n-2) \tag{39}$$

(32), (37), (38), (39)の式に形によく注意すると，これらは次のような**3重対角行列**（主対角線およびその上下の対角線以外の要素が0の行列）を係数行列に持つ，A_i, B_i, C_i $(i=1, \dots, n-1)$ に関する連立方程式となる（$\nabla_i=y_{i+1}-y_i$ と書く）。

$$
(40)\quad
\left\{
\begin{array}{lll}
A_1 & & = a\Delta_1 \\
\quad\cdots & & \cdots \\
\quad A_i + B_i + C_i & & = \nabla_i \\
① \quad B_i + 2C_i - \dfrac{\Delta_i}{\Delta_{i+1}} A_{i+1} & & = -\nabla_i \\
② \quad C_i + \dfrac{\Delta_i}{\Delta_{i+1}} A_{i+1} - \left(\dfrac{\Delta_i}{\Delta_{i+1}}\right)^2 B_{i+1} & & = \nabla_i \\
\qquad\qquad \cdots & & \cdots \\
\qquad\qquad A_{n-1} + B_{n-1} + C_{n-1} & & = \nabla_{n-1} \\
③ \qquad\qquad B_{n-1} + 2C_{n-1} & & = b\Delta_{n-1} - \nabla_{n-1}
\end{array}
\right.
$$

中央の3つの式が$i=1, \ldots, n-2$の$(n-2)$組

この方程式の1番最後の式は，(37)の後半の式から(32)の$i=n-1$のときの式を引いたものである（①, ②などは後の説明で引用するためのもの)。

(40)は上から順番に以下のような手順で変形することによって，次のような**上3角行列**（主対角線より下の要素が0の行列）を係数行列に持つ，$C_1, A_i, B_i, C_i (i=2, \ldots, n-1)$に関する連立方程式に構成しなおすことができる。

$$
(41)\quad
\left\{
\begin{array}{ll}
C_1 - \dfrac{\Delta_1}{\Delta_2} A_2 & = -\nabla_1 - (\nabla_1 - a\Delta_1) \\
\quad\cdots & \cdots \\
④ \quad C_i + \alpha_i A_{i+1} & = u_i \\
⑤ \qquad\quad A_{i+1} + \beta_i B_{i+1} & = v_i \\
⑥ \qquad\qquad\quad B_{i+1} + \gamma_i C_{i+1} & = w_i \\
\qquad\qquad \cdots & \cdots \\
⑦ \qquad\qquad\qquad\qquad C_{n-1} = \dfrac{b\Delta_{n-1} - \nabla_{n-1} - w_{n-2}}{2 - \gamma_{n-2}} &
\end{array}
\right.
$$

④, ⑤, ⑥の式が$i=1, \ldots, n-2$の$(n-2)$組

（手順 1） 方程式(40)においてA_1は$A_1 = a\Delta_1$とすでに求まっているので，(40)で2番目に位置する方程式

$$A_1 + B_1 + C_1 = \nabla_1$$

は

$$B_1 + C_1 = \nabla_1 - a\Delta_1$$

とすることができる。この式を(40)の3番目に位置する方程式

$$B_1 + 2C_1 - \frac{\Delta_1}{\Delta_2} A_2 = -\nabla_1$$

から引くと(41)の最初の式，すなわち④に相当する式が得られる。すなわち(41)において

$$\alpha_1 = -\frac{\Delta_1}{\Delta_2}, \quad u_1 = -\nabla_1 - (\nabla_1 - a\Delta_1) \tag{42}$$

である。

（手順 2） (40)の②の形の式から（手順 1）で得られた(41)の④の形の式を引き，さらに A_{i+1} の係数を 1 にする作業を行って(41)の⑤の形の式を得る。すなわち

$$\frac{\Delta_i}{\Delta_{i+1}}A_{i+1}-\left(\frac{\Delta_i}{\Delta_{i+1}}\right)^2 B_{i+1}-\alpha_i A_{i+1}=\nabla_i-u_i$$

において，A_{i+1} の係数を 1 にする作業を行って

$$\beta_i=-\left(\frac{\Delta_i}{\Delta_{i+1}}\right)^2\Big/\left(\frac{\Delta_i}{\Delta_{i+1}}-\alpha_i\right)$$

$$v_i=(\nabla_i-u_i)\Big/\left(\frac{\Delta_i}{\Delta_{i+1}}-\alpha_i\right) \tag{43}$$

が得られる。

（手順 3） (40)の②の式の次に位置する式

$$A_{i+1}+B_{i+1}+C_{i+1}=\nabla_{i+1}$$

から（手順 2）で得られた(41)の⑤の形の式を引き，さらに B_{i+1} の係数を 1 にする作業を行って(41)の⑥の形の式を得る。すなわち

$$B_{i+1}+C_{i+1}-\beta_i B_{i+1}=\nabla_{i+1}-v_i$$

において，B_{i+1} の係数を 1 にする作業を行って

$$\gamma_i=\frac{1}{1-\beta_i},\quad w_i=\frac{\nabla_{i+1}-v_i}{1-\beta_i} \tag{44}$$

が得られる。

（手順 4） γ_i, w_i の i が $n-2$ なら（手順 6）へ ((40)の式は上から順番に変形していることに注意せよ)。

（手順 5） i を 1 増やして(40)の①の形の式から（手順 3）で得られた⑥の形の式を引き，さらに C_i の係数を 1 にする作業を行って(41)の④の形の式を得る。すなわち

$$2C_i-\frac{\Delta_i}{\Delta_{i+1}}A_{i+1}-\gamma_{i-1}C_i=-\nabla_i-w_{i-1}$$

において，C_iの係数を 1 にする作業を行って

$$\alpha_i=-\frac{\Delta_i}{\Delta_{i+1}}\Big/(2-\gamma_{i-1})$$

$$u_i=\frac{-\nabla_i-w_{i-1}}{2-\gamma_{i-1}} \tag{45}$$

が得られる。（手順 2）へ。

（手順 6） (40)の③の式から(41)の⑥の形の式を引いて(41)の⑦を得る。すなわち

$$2C_{n-1}-\gamma_{n-2}C_{n-1}=b\Delta_{n-1}-\nabla_{n-1}-w_{n-2}$$

から⑦が得られる。

このように(40)の3重対角の形の連立方程式は上の方から順番に(41)の上3角の形の連立方程式に構成しなおすことができる。一たん(41)の形になれば C_{n-1} の値が決まるので，今度は**後退代入**を行うことによって下の方から順番に他の未知数を求めていくことができる。最後に B_1 は後退代入を行って得られた C_1 と，最初にすでに求まっている A_1 とから求められる。

以上で係数 $A_i, B_i, C_i\ (i=1, \ldots, n-1)$ が決まるので，当初の目的であった(31)を積分すればよい。すなわち

$$\begin{aligned}\int_{x_1}^{x_n} f(x)dx &= \sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} f_i(x)dx \\ &= \sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}}\left(y_i+\frac{A_i}{\Delta_i}(x-x_i)+\frac{B_i}{\Delta_i^2}(x-x_i)^2+\frac{C_i}{\Delta_i^3}(x-x_i)^3\right)dx \\ &= \sum_{i=1}^{n-1}\left[y_i x+\frac{A_i}{2\Delta_i}(x-x_i)^2+\frac{B_i}{3\Delta_i^2}(x-x_i)^3+\frac{C_i}{4\Delta_i^3}(x-x_i)^4\right]_{x_i}^{x_{i+1}} \\ &= \sum_{i=1}^{n-1}\Delta_i\left(y_i+\frac{A_i}{2}+\frac{B_i}{3}+\frac{C_i}{4}\right) \qquad (46)\end{aligned}$$

この(46)が**スプライン関数による数値積分の公式**である。

例題 3-7 スプライン関数による数値積分

データの組 $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ が与えられたとき，これらを通るスプライン関数 $f(x)$ 構成し，$\int_{x_1}^{x_n} f(x)dx$ を求める関数を作る。

●主プログラムで適当に用意しておくもの

const *n*=8; *n*1=7; *n*2=6;　　n はデータの個数，$n1=n-1$，$n2=n-2$。

type *t*1=***array***[1..*n*] ***of*** *real*;

var *x*, *y* : *t*1;　　$(x_1, y_1), \ldots, (x_n, y_n)$ のための1次元配列。

　a, *b* : *real*;　　端点における微分係数。

●関数

function *spline* (***var*** *x*, *y* : *t*1; *a*, *b* : *real*) : *real*;

{データの組 $(x_1, y_1), \ldots, (x_n, y_n)$ および x_1, x_n における1次微分係数 a, b が与えられたとき，これらを通るスプライン関数 $f(x)$ を構成し，$\int_{x_1}^{x_n} f(x)dx$ を求める}

var *ai*, *bi*, *ci* : ***array***[1..*n*1] ***of*** *real*;　　係数 A_i, B_i, C_i のための配列。

　alpha, *beta*, *u*, *v* : ***array***[1..*n*2] ***of*** *real*;　　$\alpha_i, \beta_i, u_i, v_i$ のための配列。

　gamma, *w* : ***array***[0..*n*2] ***of*** *real*;　ループの調整のため γ_0, w_0 からとっておく。

　dx, *dxd*, *dy*, *dyd* : *real*;　　$\Delta_i, \Delta_{i+1}, \nabla_i, \nabla_{i+1}$ に相当するもの。

　i : *integer*;

　s : *real*;

begin

```
gamma[0] := 1; w[0] := (y[2]-y[1])-a*(x[2]-x[1]);
```

$r_0=1$, $w_0=\nabla_1-a\Delta_1$ という初期化。これは(45)の式を最初から使うためである。

```
for i := 1 to n2 do                      前進しながら上3角化を行う。
  begin
    dx := x[i+1]-x[i]; dxd := x[i+2]-x[i+1];        Δi, Δi+1 の計算。
    dy := y[i+1]-y[i]; dyd := y[i+2]-y[i+1];        ∇i, ∇i+1 の計算。
    alpha[i] := -dx/dxd/(2-gamma[i-1]);
    u[i] := (-dy-w[i-1])/(2-gamma[i-1]);
                                             式(45)より αi, ui を求める。
    beta[i] := -(dx/dxd)*(dx/dxd)/(dx/dxd-alpha[i]);
    v[i] := (dy-u[i])/(dx/dxd-alpha[i]);       式(43)より βi, vi を求める。
    gamma[i] := 1/(1-beta[i]);
    w[i] := (dyd-v[i])/(1-beta[i])             式(44)より γi, wi を求める。
  end;                                   ここまでで上3角化が終わった。
ci[n-1] := (b*(x[n]-x[n-1])-(y[n]-y[n-1])-w[n-2])/
           (2-gamma[n-2]);                          Cn-1 を求める。
for i := n-2 downto 1 do       後退代入しながら Ai, Bi, Ci を求めていく。
  begin
    bi[i+1] := w[i]-gamma[i]*ci[i+1];
    ai[i+1] := v[i]-beta[i]*bi[i+1];
    ci[i] := u[i]-alpha[i]*ai[i+1]
  end;               これで Cn-1, Bn-1, An-1, ..., C2, B2, A2, C1 が求まった。
    ai[1] := a*(x[2]-x[1]);                            A1 は aΔ1 であった。
    bi[1] := (y[2]-y[1])-ci[1]-ai[1]; A1+B1+C1=∇1 から B1 を求める。
                       以上で係数 Ai, Bi, Ci (i=1, ..., n-1) が求まった。
                                 最後に式(46)を用いて定積分の値を求める。
s := 0;
for i := 1 to n-1 do
  s := s+(x[i+1]-x[i])*(y[i]+ai[i]/2+bi[i]/3+ci[i]/4);
spline := s
end;
```

【実行例】

$\int_0^\pi \sin x dx$ を次のようにデータを与えて求める（上段が x の値，下段が対応する $\sin x$ の値。$\sin x$ の値は真値の最下位のけたを四捨五入したものである)。

```
0      0.4      0.9      1.4      1.7      2.4      2.9      3.14
0 0.38942 0.78333 0.98545 0.99166 0.67546 0.23925 0.00159
```

$\sin x$ の導関数は $\cos x$ であるから，端点の 0 と 3.14 における微分係数はそれぞれ 1 と -1 で与える。そうすると

```
INTEGRAL BY SPLINE FUNCION = 1.9996E+00
```

と得られる（[例題 3-6] のシンプソンの 3/8 則による実行例と比較してみよ。こちらの方は等間隔データではなく，また対応する関数値も近似値で与えており，かつ 7 等分に相当するデータを与えているにもかかわらず，結果はシンプソンの 3/8 則による 9 等分計算の近似値と同じ精度である）。

3.4 常微分方程式の解法

微分方程式とは導関数を含む方程式のことである。たとえば

$$y'=y$$

は導関数が自分自身であるという微分方程式である。微分方程式を解くとはその方程式を満たす関数を求めることである。たとえば上の解は

$$y=ce^x \quad (c \text{ は定数})$$

である。実際この形の関数 y はすべて $y'=y$ を満たす。定数 c は y の 1 つの値が知れれば決まる。たとえば

$$y(0)=1$$

という条件があれば，$ce^0=1$ より $c=1$ となる。この条件を微分方程式の**初期条件**という。

微分方程式が n 階導関数 $y^{(n)}$ を含むとき **n 階の微分方程式**という。上の例は 1 階の微分方程式である。また微分方程式が導関数（0階導関数 y も含む）に関して 1 次式であるとき（すなわち y'^2 や y''^3 などを含まない），**線型**であるという。そうでないものを**非線型**という。上の例は線型である。

微分方程式の求められるべき関数 y が 1 変数の関数であるとき，その微分方程式を**常微分方程式**と呼ぶ。そうでなく偏導関数 $\dfrac{\partial y}{\partial x_1}$（$x_1$ 以外は定数とみなして x_1 について微分した関数。x_1 について偏微分するという），$\dfrac{\partial y}{\partial x_2}$ などを含むものを**偏微分方程式**という。上の例は結局，1 階線型常微分方程式である。

微分方程式を数値的に解くとは y の具体的な形を求めるわけではなく，適当な点での解の関数の関数値を求めることである。つまり関数値表を求めることに相当する。

以下では

$$y'=f(x,y)$$

の形の 1 階常微分方程式を扱う。高階微分方程式は 1 階微分方程式の連立方程式の形に書けるので，数値解法としては $y'=f(x,y)$ の数値解法に基本的に含まれ