Obliczanie wartości i współczynników funkcji sklejanej stopnia trzeciego z zadanymi wartościami jej pochodnej na końcach przedziału

Sebastian Maciejewski
Nr indeksu 132275

31 maja 2018

1 Zastosowanie i opis funkcji

Funkcja *clampedsplinevalue* oblicza w zadanym punkcie wartość funkcji sklejanej stopnia trzeciego, która interpoluje funkcję f p znanych wartościach $f(x_i)$ dla $i = 0, 1, ..., n$ oraz wartościach $f'(x_0)$ i $f'(x_n)$.

Procedura *clampedsplinecoeffns* oblicza współczynniki przy kolejnych potęgach x funkcji sklejanej trzeciego stopnia, tj.

$$S(x) = a_{0,i} + a_{1,i}x + a_{2,i}x^2 + a_{3,i}x^3, \qquad i = 0, 1, ..., n-1,$$

interpolującej funkcję f, dla której znane są wartości $f(x_i)$ w węzłach x_i oraz $f'(x_0)$ i $f'(x_n)$.

2 Opis metody

Wyznaczanie wartości

Funkcję sklejaną stopnia trzeciego można w każdym z przedziałów $[x_i, x_{i-1})$ przedstawić w postaci

$$S(x) = a_i + b_i t + c_i t^2 + d_i t^3 \tag{1}$$

gdzie $t = x - x_i$ dla $x \in [x_i, x_{i-1}), i = 0, 1, ..., n-1$. W celu wyznaczenia funkcji sklejanej postaci (1), która spełnia warunki

$S(x_i) = f(x_i)$ dla $i = 0, 1, ..., n-1$ oraz $S'(x_0) = f'(x_0)$ i $S'(x_n) = f'(x_n)$,

tj. określenia współczynników a_i, b_i, c_i oraz d_i dla $i = 0, 1, ..., n-1$ należy najpierw rozwiązać układ równań liniowych

$$\begin{aligned} &2M_0 + \lambda_0 M_1 = \delta_0 \\ &\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = \delta_i, \quad i = 1, 2, ..., n-1, \\ &\mu_n M_{i-1} + 2M_n = \delta_n, \end{aligned} \tag{2}$$

w którym współczynniki $\lambda_i (i = 0, 1, ..., n-1)$, $\mu_i (i = 1, 2, ..., n)$ i $\delta_i (i = 0, 1, ..., n)$ są określone następującymi wzorami:

$$
\begin{aligned}
&\lambda_0 = 1,\\
&\delta_0 = \frac{6}{h_1}\left(\frac{f(x_1)-f(x_0)}{h_1} - f'(x_0)\right),\\
&\lambda_i = \frac{h_{i+1}}{h_{i+1}+h_i},\\
&\mu_i = 1-\lambda_i,\\
&\delta_i = \frac{6}{h_{i+1}+h_i}\left(\frac{f(x_{i+1})-f(x_i)}{h_{i+1}} - \frac{f(x_i)-f(x_{i-1})}{h_i}\right), \quad i = 1,2,...,n-1,\\
&\mu_n = 1,\\
&\delta_n = \frac{6}{h_n}\left(f'(x_n) - \frac{f(x_n)-f(x_{n-1})}{h_n}\right),
\end{aligned}
$$

przy czym $h_{i+1} = x_{x+1} - x_i$. Po rozwiązaniu układu równań (2) współczynniki funkcji sklejanej wyznacza się z następujących zależności:

$$
\begin{aligned}
a_i &= f(x_i),\\
b_i &= \frac{f(x_{i+1})-f(x_i)}{h_{i+1}} - \frac{2M_i+M_{i+1}}{6}h_{i+1},\\
c_i &= \frac{M_i}{2},\\
d_i &= \frac{M_{i+1}-M_i}{6h_{i+1}}.
\end{aligned}
$$

W funkcji *clampedsplinevalue* układ równań liniowych (2) rozwiązuje się metodą Crouta, a następnie określa się przedział $[x_i, x_{i+1})$ zawierający dany punkt x i dla tego przedziału wyznacza się współczynniki a_i, b_i, c_i oraz d_i. Następnie z wzoru (1) oblicza się wartość funkcji sklejanej.

Wyznaczanie współczynników

Współczynniki obliczane przez procedurę *clampedsplinecoeffns*, tj. $a_{0,i}, a_{1,i}, a_{2,i}$ oraz $a_{3,i}$, są związane ze współczynnikami a_i, b_i, c_i oraz d_i funkcji sklejanej S zapisanej w postaci (1) następującymi zależnościami:

$$
\begin{aligned}
a_{0,i} &= a_i - b_i x_i + c_i x_i^2 - d_i x_i^3,\\
a_{1,i} &= b_i - 2c_i x_i + 3d_i x_i^2,\\
a_{2,i} &= c_i - 3d_i x_i,\\
a_{3,i} &= d_i.
\end{aligned}
$$

W celu otrzymania współczynników funkcji S zapisanej w postaci (1) należy skorzystać z następujących wzorów:

$$
\begin{aligned}
a_i &= a_{0,i} + a_{1,i}x_i + a_{2,i}x_i^2 + a_{3,i}x_i^3,\\
b_i &= a_{1,i} + 2a_{2,i}x_i + 3a_{3,i}x_i^2,\\
c_i &= a_{2,i} + 3a_{3,i}x_i,\\
d_i &= a_{3,i}.
\end{aligned}
$$

3 Wywołanie procedury

Wyznaczanie wartości

intervalclampedsplinevalue (n, x, f, f1x0, f1xn, xx, st)

Wyznaczanie współczynników

intervalclampedsplinecoeffns (n, x, f, f1x0, f1xn, a, st)

4 Dane

n – liczba węzłów interpolacji minus 1 (węzły są ponumerowane od 0 do n),

x – tablica zawierająca wartości węzłów,

f – tablica zawierająca wartości interpolowanej funkcji w węzłach,

$f1x0$ – wartość $f'(x_0)$,

$f1xn$ – wartość $f'(x_n)$,

xx – punkt, w którym należy obliczyć wartość naturalnej funkcji sklejanej stopnia trzeciego.

5 Wyniki

Wyznaczanie wartości

clampedsplinevalue (n,x,f,f1x0,f1xn,xx,st) - wartość funkcji sklejanej stopnia trzeciego w punkcie xx.

Wyznaczanie współczynników

a – tablica współczynników funkcji sklejanej (element $a[k,i]$ zawiera wartość współczynnika przy $x^k (k = 0, 1, 2, 3)$ dla przedziału $[x_i, x_{i+1}),\ \ i = 0, 1, ..., n-1$).

6 Inne parametry

st – zmienna, której w wyniku wykonania funkcji zostanie przypisana jedna z następujących wartości:

1, jeżeli $n < 1$,
2, gdy istnieją równe wartości $x[i]$ i $x[j]$ dla $i \neq j (i, j = 0, 1, ..., n)$,
3, jeśli $xx < x[0]$ lub $xx > x[n]$,
0, w przeciwnym przypadku.

Uwaga: Jeżeli $st \neq 0$, to wartość funkcji *clampedsplinevalue* i elementy tablicy a w funkcji *clampedsplinecoeffns* nie są obliczane.

7 Typy parametrów

Wyznaczanie wartości

Integer: n, st
interval: $f1x0, f1xn, xx$
Ivector: x, f

Wyznaczanie współczynników

Integer: n, st
interval: $f1x0, f1xn$
Ivector: x, f
Imatrix: a

8 Identyfikatory nielokalne

Ivector – nazwa typu tablicowego $[q_0 ... q_n]$ o elementach typu *interval*

Imatrix – nazwa typu tablicowego $[0..3, q_0 ... q_{n-1}]$ o elementach typu *interval*

9 Funkcje i procedury

```
procedure intervalclampedsplinecoeffns (n          : Integer;
                                        x,f        : Ivector;
                                        f1x0,f1xn  : interval;
                                        var a      : Imatrix;
                                        var st     : Integer);

var i,k          : Integer;
    u,v,y,z,xi   : interval;
    d            : Ivector;
    b            : Ivector1;
    c            : Ivector2;
begin
SetLength(b,n);
SetLength(c,n);
SetLength(d,n);

  if n<1
    then st:=1
    else begin
           st:=0;
           i:=-1;
           repeat
             i:=i+1;
             for k:=i+1 to n do
               if compare_equal(x[i],x[k])
                 then st:=2
           until (i=n-1) or (st=2)
         end;
  if st=0
    then begin
           b[0]:=1;
           u:=x[1]-x[0];
           d[0]:=6*((f[1]-f[0])/u-f1x0)/u;
           c[n]:=1;
           u:=x[n]-x[n-1];
           d[n]:=6*(f1xn-(f[n]-f[n-1])/u)/u;
           for i:=1 to n-1 do
             begin
               z:=x[i];
               y:=x[i+1]-z;
               z:=z-x[i-1];
               u:=f[i];
               b[i]:=y/(y+z);
               c[i]:=1-b[i];
               d[i]:=6*((f[i+1]-u)/y-(u-f[i-1])/z)/(y+z)
             end;
           u:=2;
           i:=-1;
           y:=d[0]/u;
           d[0]:=y;
           repeat
             i:=i+1;
             z:=b[i]/u;
             b[i]:=z;
             u:=2-z*c[i+1];
             y:=(d[i+1]-y*c[i+1])/u;
             d[i+1]:=y
           until i=n-1;
           u:=d[n];
           for i:=n-1 downto 0 do
             begin
               u:=d[i]-u*b[i];
               d[i]:=u
```

```
            end;
          for i:=0 to n-1 do
            begin
              u:=f[i];
              xi:=x[i];
              z:=x[i+1]-xi;
              y:=d[i];
              v:=(f[i+1]-u)/z-(2*y+d[i+1])*z/6;
              z:=(d[i+1]-y)/(6*z);
              y:=y/2;
              a[0,i]:=((-z*xi+y)*xi-v)*xi+u;
              u:=3*z*xi;
              a[1,i]:=(u-2*y)*xi+v;
              a[2,i]:=y-u;
              a[3,i]:=z
            end
        end
end;


function intervalclampedsplinevalue (n             : Integer;
                                     x,f           : Ivector;
                                     f1x0,f1xn,xx  : interval;
                                     var st        : Integer) : interval;

var i,k    : Integer;
    u,y,z  : interval;
    found  : Boolean;
    a      : array [0..3] of interval;
    d      : Ivector;
    b      : Ivector1;
    c      : Ivector2;
begin
SetLength(b,n);
SetLength(c,n);
SetLength(d,n);
  if n<1
    then st:=1
    else if (compare_less(xx,x[0])) or compare_less(x[n],xx)
           then st:=3
           else begin
                  st:=0;
                  i:=-1;
                  repeat
                    i:=i+1;
                    for k:=i+1 to n do
                      if compare_equal(x[i],x[k])
                        then st:=2
                  until (i=n-1) or (st=2)
                end;
  if st=0
    then begin
           b[0]:=1;
           u:=x[1]-x[0];
           d[0]:=6*((f[1]-f[0])/u-f1x0)/u;
           c[n]:=1;
           u:=x[n]-x[n-1];
           d[n]:=6*(f1xn-(f[n]-f[n-1])/u)/u;
           for i:=1 to n-1 do
             begin
               z:=x[i];
               y:=x[i+1]-z;
               z:=z-x[i-1];
               u:=f[i];


               b[i]:=y/(y+z);
               c[i]:=1-b[i];
               d[i]:=6*((f[i+1]-u)/y-(u-f[i-1])/z)/(y+z)
             end;
           u:=2;
           i:=-1;
           y:=d[0]/u;
```

```
          d[0]:=y;
          repeat
            i:=i+1;
            z:=b[i]/u;
            b[i]:=z;
            u:=2-z*c[i+1];
            y:=(d[i+1]-y*c[i+1])/u;
            d[i+1]:=y
          until i=n-1;
          u:=d[n];
          for i:=n-1 downto 0 do
            begin
              u:=d[i]-u*b[i];
              d[i]:=u
            end;
          found:=False;
          i:=-1;
          repeat
            i:=i+1;
            if (compare_less_or_equal(x[i],xx) and
            (compare_less_or_equal(xx,x[i+1])))
              then found:=True
          until found;
          y:=x[i+1]-x[i];
          z:=d[i+1];
          u:=d[i];
          a[0]:=f[i];
          a[1]:=(f[i+1]-f[i])/y-(2*u+z)*y/6;
          a[2]:=u/2;
          a[3]:=(z-u)/(6*y);
          y:=a[3];
          z:=xx-x[i];
          for i:=2 downto 0 do
            y:=y*z+a[i];
          intervalclampedsplinevalue:=y
        end
end;
```

10 Przykłady

Błędne przykłady

```
Dane:
n=1, x[0]=2, x[1]=3, f[0]=0, f[1]=1, f1x0=1, f1xn=1, xx=0,5

Wyniki:
st = 3


Dane:
n=1, x[0]=(1;2), x[1]=(1;2), f[0]=(3;4), f[1]=(3;4), f1x0=(5;6), f1xn=(5;6)

Wyniki:
st = 2
```

Zwykła arytmetyka

```
Obliczanie wartosci:
Dane:
n=1, x[0]=0, x[1]=1, f[0]=0, f[1]=1, f1x0=1, f1xn=1, xx=0,5

Wyniki:
Wynik=5.00000000000000E-0001,
st=0


Obliczanie wspolczynnikow:
Dane:
n=1, x[0]=0, x[1]=1, f[0]=0, f[1]=1,
f1x0=1, f1xn=1

Wyniki:
a[0,0]=0.0000000000000000E+0000 a[1,0]=1.000000000000000E+0000
a[2,0]=0.0000000000000000E+0000 a[3,0]=0.000000000000000E+0000
st=0
```

Arytmetyka przedziałowa

```
Obliczanie wartosci:
Dane:
n=1, x[0]=(0;0), x[1]=(1;1), f[0]=(0;0), f[1]=(1;1), f1x0=(1;1), f1xn=(1;1,1),
xx=(0,5;0,6)

Wyniki:
Wynik=(4.3666666666666666E-0001, 6.6160000000000000E-0001),
Szerokosc przedzialu = 2,249E-01,
st=0


Obliczanie wspolczynnikow:
Dane:
n=1, x[0]=(0;0), x[1]=(1;1,1), f[0]=(0;0), f[1]=(1;1),
f1x0=(1;1), f1xn=(1;1)

Wyniki:
a[0,0]=(0.0000000000000000E+0000, 0.0000000000000000E+0000)
Szerokosc przedzialu = 0.000E+00
a[1,0]=(8.0909090909090909E-0001, 1.2000000000000001E+0000)
Szerokosc przedzialu = 3.909E-01
a[2,0]=(-2.7272727272727273E-0001, 0.0000000000000000E+0000)
Szerokosc przedzialu = 2.727E-01
a[3,0]=(0.000000000000000E+0000, 1.8181818181818182E-0001)
Szerokosc przedzialu = 1.818E-01
st=0
```