5.10 Kubiska splinefunktioner

Definition 5.10.1 *Låt* $a = x_1 < x_2 < \ldots < x_n = b$ *vara givna punkter,* **noder**. *Funktionen* s *sägs vara en* **kubisk splinefunktion** *på intervallet* $[a, b]$ *om*

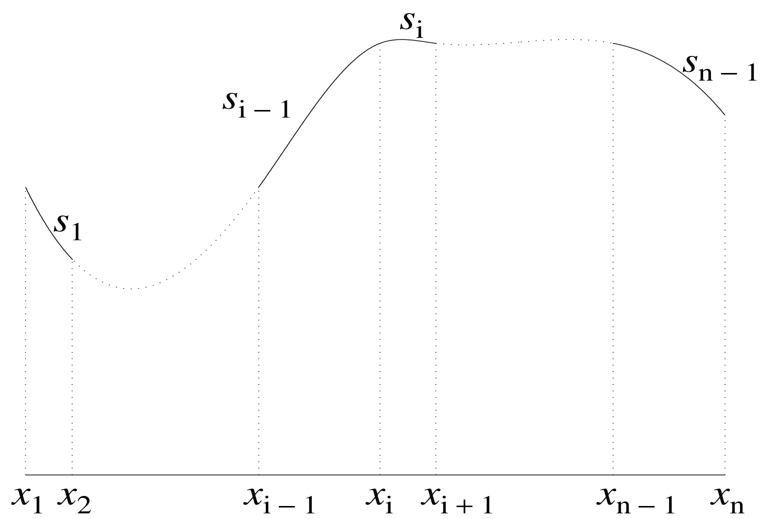
s, s' *och* s'' *är kontinuerliga på* $[a, b]$,

s *är ett polynom av grad* ≤ 3 *på varje delintervall* $[x_i, x_{i+1}]$, $i = 1, \ldots, n-1$.

Splinefunktionen s är sammansatt av $n-1$ polynom:

$$s(x) = s_i(x), \qquad x_i \leq x \leq x_{i+1},$$

se Figur 5.8.



Figur 5.8: Kubisk splinefunktion.

Den sammansatta funktionen s är kontinuerlig och har kontinuerlig derivata, om för $i = 1, \ldots, n-1$ gäller att

$$s_i(x_i) = y_i \quad \text{och} \quad s_i(x_{i+1}) = y_{i+1}, \tag{5.4}$$
$$s_i'(x_i) = y_i' \quad \text{och} \quad s_i'(x_{i+1}) = y_{i+1}'. \tag{5.5}$$

Funktionen s'' är kontinuerlig, om

$$s_{i-1}''(x_i) = s_i''(x_i) \quad i = 2, \ldots, n-1 \tag{5.6}$$

Här är y_i och y_i' godtyckliga. Om ekvationerna (5.4), (5.5) och (5.6) är uppfyllda, är s en kubisk splinefunktion. Ekvationerna innebär $(n-2) + 4(n-1)$ krav. Vi förfogar över $(n-1)$ kubiska polynom s_i, har alltså $4(n-1)$ koefficienter och kan dessutom välja $y_1, \ldots, y_n$ och $y_1', \ldots, y_n'$. Vi har alltså $2n - (n-2) = n+2$ frihetsgrader kvar när väl kraven för att s skall vara en splinefunktion är tillgodosedda. För att få en **interpolerande** splinefunktion väljer vi

$$y_i = f_i \qquad i = 1, 2, \ldots, n, \tag{5.7}$$

där $f_i = f(x_i)$. De återstående båda frihetsgraderna kan utnyttjas på olika sätt. Man kan t ex arbeta med en “naturlig splinefunktion“ eller föreskriva “rätta randvillkor”, “not-a knot”-villkor eller periodiska randvillkor. Vad innebär då dessa villkor?

En träribba som fixeras i noderna fortsätter rätlinjigt utanför intervallet $[a, b]$. Det verkar alltså vara naturligt att kräva att

$s_1''(x_1) = 0,$ $s_{n-1}''(x_n) = 0,$	“naturlig” splinefunktion.

Alternativt kan man föreskriva

$s_1'(x_1) = f'(x_1),$ $s_{n-1}'(x_n) = f'(x_n),$	“rätta randvillkor”.

En splinefunktion med rätta randvillkor ger i regel en bättre approximation till f nära x_1 och x_n än en naturlig splinefunktion (såvida inte $f'' \sim 0$ nära x_1 och x_n). Om man inte känner till derivatorna av f i x_1 och x_n kan man ge "nästan rätta randvillkor" genom att ersätta derivatorna med derivatan av det kubiska polynom som interpolerar funktionsvärdena i de fyra första respektive de fyra sista noderna.

Tredje derivatan $s^{(3)}$ av en kubiska splinefunktion är konstant i varje delintervall, men gör i regel ett språng vid övergången från ett intervall till nästa. Om man kräver att

$$\boxed{\begin{array}{rl} s_1^{(3)}(x_2) = s_2^{(3)}(x_2), & \\ s_{n-2}^{(3)}(x_{n-1}) = s_{n-1}^{(3)}(x_{n-1}), & \quad \text{“not-a-knot”,} \end{array}}$$

blir $s_1(x) = s_2(x)$ och $s_{n-2}(x) = s_{n-1}(x)$.

Vid approximation av periodiska funktioner är det rimligt att kräva

$$\boxed{\begin{array}{rl} s_1'(x_1) = s_{n-1}'(x_n), & \\ s_1''(x_1) = s_{n-1}''(x_n), & \quad \text{“periodiska randvillkor”.} \end{array}}$$

Om dessa randvillkor är uppfyllda, kan den i (x_1, x_n) definierade splinefunktionen fortsättas periodiskt till vänster och till höger, och den resulterande sammansatta funktionen är åter en splinefunktion ($s(x_1) = s(x_n)$ förutsättes gälla p g a periodiciteten).

Nu bestämmer vi y_i' och koefficienterna i s_i. Vi gör ansatsen

$$s_i(x) = a_i + b_i\left(\frac{x - x_i}{h_i}\right) + c_i\left(\frac{x - x_i}{h_i}\right)^2 + d_i\left(\frac{x - x_i}{h_i}\right)^3,$$

där

$$h_i = x_{i+1} - x_i$$

betecknar avståndet mellan noderna.

Kraven i ekvationerna (5.4), (5.5) och (5.7) är uppfyllda, om för $i = 1, \ldots, n-1$ gäller att

$$\begin{aligned} a_i &= f_i, \\ a_i + b_i + c_i + d_i &= f_{i+1}, \\ b_i &= h_i y_i', \\ b_i + 2c_i + 3d_i &= h_i y_{i+1}', \end{aligned}$$

dvs

$$\begin{aligned} a_i &= f_i, \\ b_i &= h_i y_i', \\ c_i &= 3(f_{i+1} - f_i) - h_i(2y_i' + y_{i+1}'), \\ d_i &= 2(f_i - f_{i+1}) + h_i(y_i' + y_{i+1}'). \end{aligned}$$

Ekvation (5.6) kan skrivas

$$\frac{2}{h_{i-1}^2} c_{i-1} + \frac{6}{h_{i-1}^2} d_{i-1} = \frac{2}{h_i^2} c_i, \quad i = 2, \ldots, n-1.$$

Efter insättning av uttrycken för koefficienterna och förenkling ger detta

$$\frac{1}{h_{i-1}} y_{i-1}' + \left(\frac{2}{h_i} + \frac{2}{h_{i-1}} \right) y_i' + \frac{1}{h_i} y_{i+1}' = \frac{3}{h_{i-1}^2}(f_i - f_{i-1}) + \frac{3}{h_i^2}(f_{i+1} - f_i).$$

Slutligen ger multiplikation med $h_i h_{i-1}$:

$$h_i y_{i-1}' + 2\left(h_i + h_{i-1}\right) y_i' + h_{i-1} y_{i+1}' = 3\left(h_i \frac{f_i - f_{i-1}}{h_{i-1}} + h_{i-1} \frac{f_{i+1} - f_i}{h_i} \right).$$

Vi har nu $(n-2)$ ekvationer och n obekanta $y_1', y_2', \ldots, y_n'$.

En naturlig splinefunktion fås, om vi dessutom kräver att

$$\begin{aligned} s_1''(x_1) &= 2c_1 = 0, \\ s_{n-1}''(x_n) &= 2c_{n-1} + 6d_{n-1} = 0 \end{aligned}$$

eller, efter insättning av uttrycken för koefficienterna,

$$\begin{aligned} 2y_1' + y_2' &= 3\frac{f_2 - f_1}{h_1} \\ y_{n-1}' + 2y_n' &= 3\frac{f_n - f_{n-1}}{h_{n-1}}. \end{aligned}$$

Vill man i stället ha en spline med rätta randvillkor, kräver man

$$y_1' = f'(x_1) \text{ och } y_n' = f'(x_n).$$

I båda fallen får man ett ekvationssystem med en tridiagonal, diagonaldominant koefficientmatris. En sådan matris kan visas vara ickesingulär, och ekvationssystemtet kan lösas med Gausselimination utan pivotering (jfr Kapitel 8).

Vi sammanfattar:

En kubisk splinefunktion s, som interpolerar en given funktion f i punkterna $x_1, x_2, \ldots, x_n$ ges för $x_i \le x \le x_{i+1}$, $i = 1, 2, \ldots, n-1$ av

$$s_i(x) = a_i + b_i \left(\frac{x - x_i}{h_i}\right) + c_i \left(\frac{x - x_i}{h_i}\right)^2 + d_i \left(\frac{x - x_i}{h_i}\right)^3,$$

där

$$h_i = x_{i+1} - x_i.$$

För $i = 1, \ldots, n-1$ är

$$\begin{aligned} a_i &= f_i, \\ b_i &= h_i y_i', \\ c_i &= 3(f_{i+1} - f_i) - h_i(2y_i' + y_{i+1}'), \\ d_i &= 2(f_i - f_{i+1}) + h_i(y_i' + y_{i+1}'). \end{aligned}$$

Här är $f_i = f(x_i)$, och y_i', $i = 1, 2, \ldots, n$, är lösning till ekvationssystemet

$$h_i y_{i-1}' + 2\,(h_i + h_{i-1})\, y_i' + h_{i-1} y_{i+1}' = 3\left(h_i \frac{f_i - f_{i-1}}{h_{i-1}} + h_{i-1} \frac{f_{i+1} - f_i}{h_i}\right),$$

$$i = 2, \ldots, n-1,$$

med tilläggsvillkoren

$$\begin{cases} 2y_1' + y_2' = 3\,\dfrac{f_2 - f_1}{h_1}, \\ y_{n-1}' + 2y_n' = 3\,\dfrac{f_n - f_{n-1}}{h_{n-1}}, \end{cases} \qquad \text{(naturlig splinefunktion)}$$

alternativt

$$y_1' = f'(x_1) \text{ och } y_n' = f'(x_n), \qquad \text{(rätta randvillkor).}$$

För fallet ekvidistanta noder och interpolation med rätta randvillkor, kan splinefunktionens derivator y_i' beräknas med följande funktion:

```
function yprim=splint(x,f,fprim1,fprim2)
 %vektorn yprim innehåller värdet i noderna
 %av första derivatan av en splinefunktion,
 %som interpolerar funktionen f med rätta randvillkor.
 %
 % x: vektor med ekvidistanta noder.
 % fprim1: derivatan av f i första  noden.
 % fprim2: derivatan av f i sista  noden.
   n=length(x);
   h=x(2)-x(1);

   %koefficientmatrisen
   a=4.*eye(n-2)+diag(ones(n-3,1),1)+diag(ones(n-3,1),-1);

   %högerledet
   for i=1:n-2
     hl(i)=3/h*(feval(f,x(i+2))-feval(f,x(i)));
   end
   yprim(1)=fprim1;
   yprim(n)=fprim2;
   hl(1)=hl(1)-yprim(1);
   hl(n-2)=hl(n-2)-yprim(n);

   yprim(2:n-1)=a\hl';
```

När derivatorna har beräknats, kan följande Matlabfunktion användas för att rita de olika segmenten $s_i - f$ av felfunktionen. I det aktuella intervallet, $x_i \leq x \leq x_{i+1}$, arbetar man med en lokal parameter $u = (x - x_i)/h_i$, som varierar mellan 0 och 1.

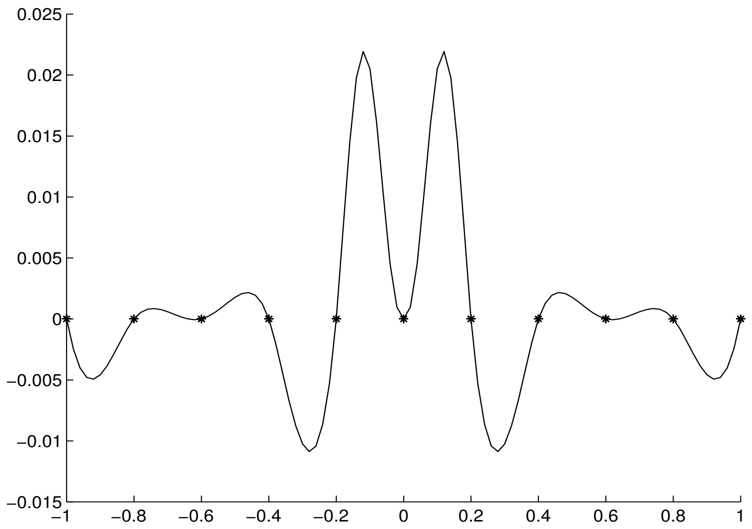
```
function y=approx(f,i,x,yprim);
 %beräknar  det i:te segmentet av
 %den interpolerande  splinefunktionen
 %till funktionen f och
 %ritar motsvarande segment av felfunktionen.
```

```
%vektorn x innehåller de ekvidistanta noderna
%vektorn yprim innehåller värdet av första derivatan
%av splinefunktionen  i noderna
  h=x(2)-x(1);
  ai=feval(f,x(i));
  bi=h*yprim(i);
  ci=3*(feval(f,x(i+1))-feval(f,x(i)));
  ci=ci-h*(2*yprim(i)+yprim(i+1));
  di=2*(feval(f,x(i))-feval(f,x(i+1)));
  di=di+h*(yprim(i)+yprim(i+1));

  %lokal parameter u
  u=0:0.1:1;
  y=((di.*u+ci).*u+bi).*u+ai;

  xx=x(i):0.1*h:x(i+1);
  plot(xx,y-feval(f,xx))
```

Exempel Låt oss interpolera $f(x) = 1/(1+25x^2)$ med en kubisk splinefunktion. Vi väljer ekvidistanta noder $x_i = -1+0.2(i-1)$, $i = 1, 2, \ldots, 11$ och rätta randvillkor, dvs $y_1' = f'(-1) = 50/26^2$ och $y_n' = f'(1) = -50/26^2$.

```
function y=f(x)
  y=1./(1+25.*x.*x);

 >>hold on
 >>x=-1:0.2:1;
 >>yprim=splint(x,@f,50/(26*26),-50/(26*26));
 >>for i=1:10
      approx(@f,i,x,yprim);
   end;
```

Resultatet visas i Figur 5.9, där även noderna har markerats med stjärna. Vi får god överensstämmelse mellan f och den approximerande splinefunktionen s.



Figur 5.9: Felfunktionen då $1/(1+25x^2)$ interpoleras med en kubisk splinefunktion.

En kubisk splinefunktion med noder $x_1 < x_2 < \ldots < x_n$ kan alternativt uttryckas med hjälp av **kubiska B-splinefunktioner,** $B_i^3(x)$, som basfunktioner:

$$s(x) = \sum_i c_i B_i^3(x).$$

Då många standardprogram utnyttjar en sådan framställning, är det bra att känna till något om B-splinefunktioner.

Funktionen B_i^3 är en kubisk splinefunktion definierad på hela reella axeln, men funktionen är skild från noll endast i intervallet (x_i, x_{i+4}). Man säger, att funktionen har sitt **stöd** i detta intervall (jfr Figur 5.10.)

Funktioner med litet stöd är mycket användbara i praktiska beräkningar. Låt oss ge ett exempel. Då kubiska B-splinefunktioner är skilda från noll endast i fyra intervall bildade av successiva noder, är högst fyra av dem skilda från noll för ett givet $x-$värde (jfr Figur 5.11). Det betyder, att $\sum_i c_i B_i^3(x)$ för ett visst $x-$värde innehåller högst fyra termer. Om vi alltså t ex kräver att s skall interpolera vissa funktionsvärden, kan motsvarande c_i bestämmas ur ett ekvationssystem med bandmatris.

Vi såg tidigare att det finns basfunktioner för **lineära** splinefunktioner som har stöd endast i **två** angränsande nodintervall. Av beviset av följande sats framgår, att minsta möjliga stöd för en **kubisk** splinefunktion, som