Approximation

Lösungshinweise

Gerhard Opfer SS 2000

Bogen 9: Ausgabetermin: am 8. Juni 2000

Abgabetermin: am 22. Juni 2000 vor den Übungen

Präsenzaufgaben:

Aufgabe P25. Schreiben Sie bei beliebiger Knotenvorgabe die B-Splines vom Grad Eins und Zwei explizit als Linearkombination von Indikatorfunktionen hin.

Lösung.

$$B_{j2} = \omega_{j,2}X_j + (1-\omega_{j+1,2})X_{j+1},$$
$$B_{j3} = \omega_{j,2}\omega_{j,3}X_j + \big(\omega_{j,3}(1-\omega_{j+1,2}) + \omega_{j+1,2}(1-\omega_{j+1,3})\big)X_{j+1} + (1-\omega_{j+1,3})(1-\omega_{j+2,2})X_{j+2}.$$

Aufgabe P26. Wie lautet die B-Spline-Darstellung linearer Polynome? Wie gewinnt man diese Darstellung?

Lösung. $p = \sum_j p(t_j^*)B_{jk}$ für alle $p \in \Pi_1$ mit $t_j^* := (t_{j+1} + t_{j+2} + \ldots + t_{j+k-1})/(k-1)$. Durch Anwendung der Darstellungsformeln (9.3) und (9.4).

Aufgabe P27. Wenn viele Bernsteinpolynome in eine Zeichnung gezeichnet werden, welche Kurve bildet dann die sichtbare Einhüllende. Im Bild 9.6 sind alle 101 Polynome vom Grad 100 gezeichnet.

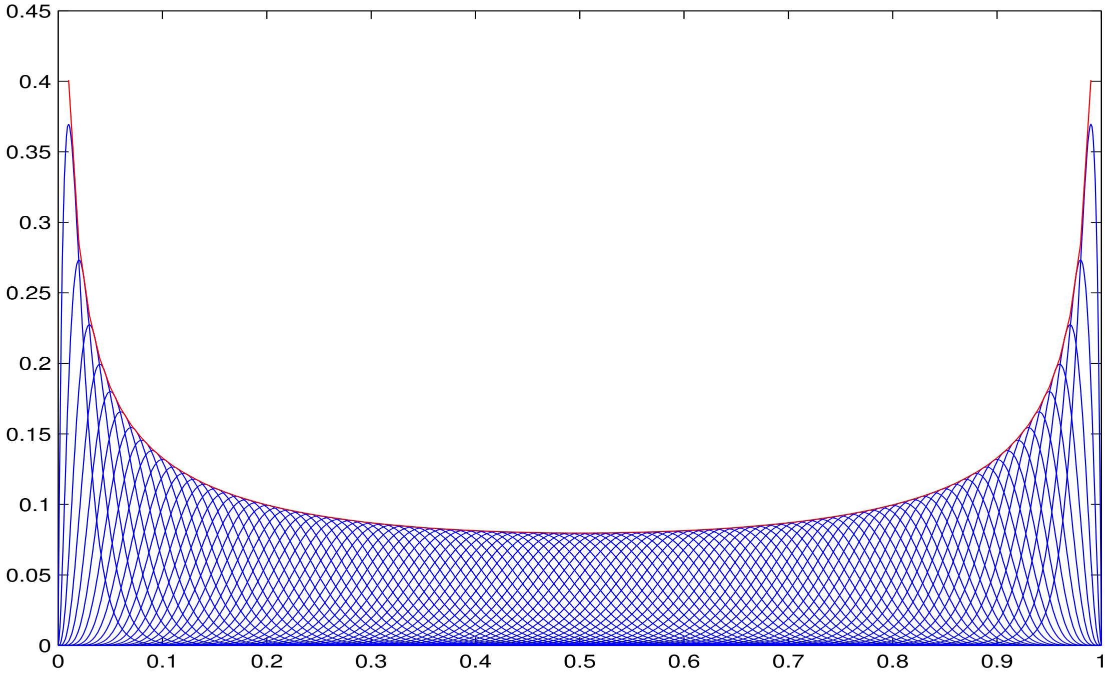

Abb. L9.1. Bernsteinpolynome vom Grad 100 und Hüllkurve (rot)

Lösung. Sei

(L9.1) $$B_j^n(t) := \binom{n}{j} t^j (1-t)^{n-j},\ j = 0, 1, \ldots, n$$

das j-te Bernstein-Polynom vom Grad n. Das Polynom B_j^n hat genau ein Maximum bei $t_j = j/n$. Das Maximum hat offensichtlich den Wert

$$w_j^n := \max_{t\in[0,1]} B_j^n(t) = B_j^n(j/n) = \binom{n}{j}(j/n)^j(1-j/n)^{n-j}.$$

Benutzen wir für $\binom{n}{j}$ Sterlings Formel, nämlich $k! \approx \sqrt{2\pi}k^{k+\frac{1}{2}}\exp(-k)$ (ABRAMOWITZ & STEGUN, Formel 6.1.38, S. 257), so ist

$$\text{(L9.2)}\qquad w_j^n \approx \tilde{w}_j^n := \frac{1}{\sqrt{2\pi}}\frac{\sqrt{n}}{\sqrt{j}\sqrt{n-j}} = \frac{1}{\sqrt{2n\pi}}\frac{1}{\sqrt{x(1-x)}}, \quad x = j/n, \quad 1 \le j \le n-1.$$

Zeichnet man $(j/n, \tilde{w}_j^n)$, $1 \le j \le n-1$ in die bereits vorhandene Zeichnung der Bernsteinpolynome Abb. 9.6 ein, so stimmt diese Kurve optisch mit der Hüllkurve total überein (s. Abb. L9.1).

Hausaufgaben:

Aufgabe 17. (a) Zeichnen Sie den zu den Knoten $0, 2, 5, 10, 14, 14, 16$ gehörenden quintischen B-Spline (also Ordnung sechs) und (b) die daran beteiligten Polynomstücke. Stellen Sie dazu ein geeignetes Programm her, das in der Lage ist, beliebige B-Splines mit vorgegebener Ordnung und Knotenfolge auszurechnen und zu zeichnen. Teilen Sie (c) mit, wie lange Ihr Rechner daran gerechnet hat und welchen Rechnertyp Sie benutzt haben: Name des Betriebssystems, Spezifikation des Prozessors. Geben Sie auch das Programm ab.

Lösung. Das Bild sieht so aus:

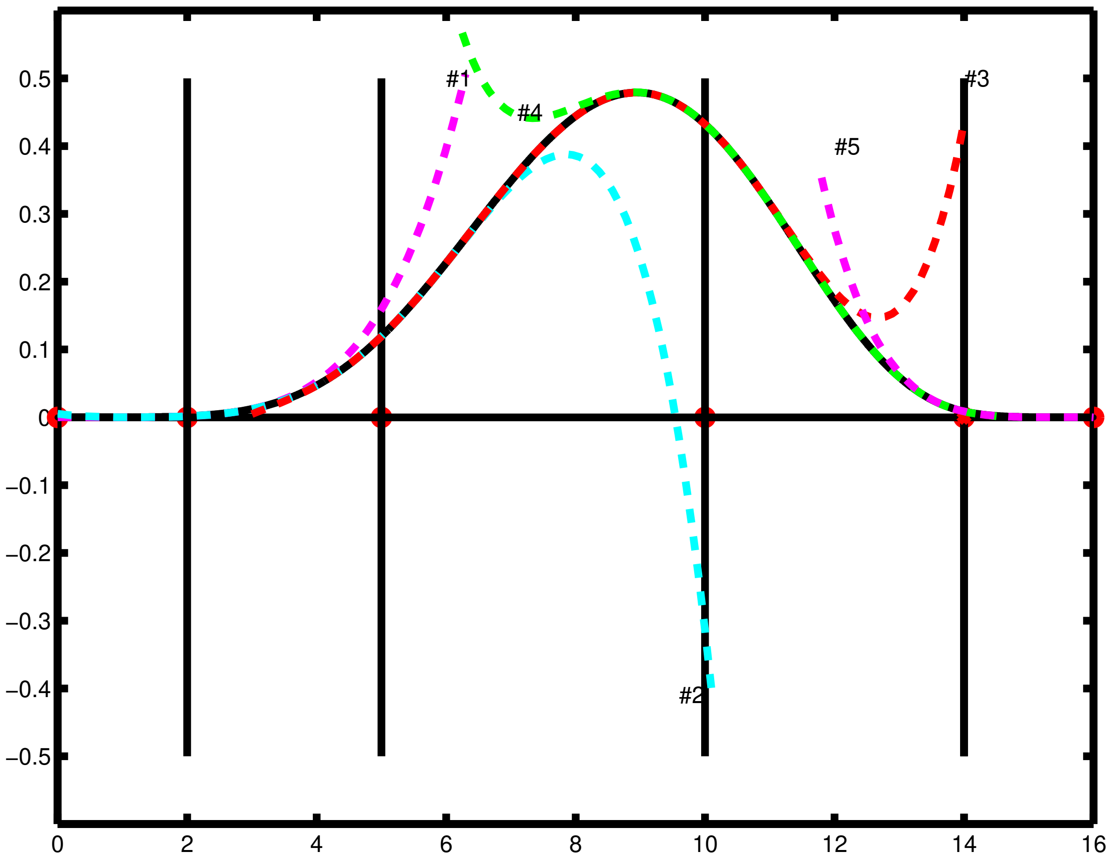

Abb. L9.2. Quintischer B-Spline und Polynomanteile
Das Programm könnte so aussehen:

Programm L9.3. B-Splines der Ordnung k mit Rekursion

```
%Alle Splines k. Ordnung (Grad k-1) sollen berechnet und gezeichnet werden.
%Wegen der Zeichnungen ist das Programm im Moment so gemacht, da\ss\ nur ganz-
%zahlige Knoten zugelassen sind und alle Funktionswerte simultan ausgerechnet
%werden. Will man nur f\"ur einen einzigen Funktionswert
%die B-Splines ausrechnen, k\"onnte das Programm vereinfacht werden.
%Programm: Gerhard Opfer, Juni 2000.
%Quelle: Carl de Boor: B(asic)-Spline Basics, ftp-server Madison, 1996.
%==========================================================================
%Vorbesetzungen
clear
Ordnung=6; %vorgegebene Ordnung
t=[0,2,5,10,14,14,16]; %gegebene Knotenfolge
if length(t)<2, error('Zu wenig Knoten'); end; %error
talt=t;
t=sort(t);
if talt  ~= t
  disp('Knoten nicht aufsteigend geordnet, Umordnung vorgenommen!');
end; %if
talt=[];
if t ~= round(t)
  error('Knoten nicht ganzzahlig');
end; %if
%Wir vergr\"o\ss ern (oder verkleinern) ggf. die Knotenfolge an beiden Enden
%so, da\ss\ die beiden Endknoten die gleichen Vielfachheiten k=Ordnung haben.
%Beginnt man also nur mit zwei Knoten, so erh\"alt man die Bernstein-Polynome.
%Ist i die Anzahl der inneren Knoten (mehrfache mehrfach gez\"ahlt),
%so ist die Gesamtzahl i+2k. Jeder Spline braucht k+1 Knoten
li=t(1);re=t(length(t));
links=find(t==li);
rechts=find(t==re);
t([links,rechts])=[];
rnd=ones(1,Ordnung);
t=[li*rnd,t,re*rnd];
lenge=length(t);
h=100;
l=h*(t(length(t))-t(1))+1;
tt=linspace(t(1),t(length(t)),l);
B=zeros(lenge-1,l);
n(1)=1;
%==========================================================================
tic;  %Die Stoppuhr wird gestartet.
%==========================================================================
%Zuerst st\"uckweise konstante Splines
for j=1:lenge-1
  n(j+1)=n(j)+(t(j+1)-t(j))*h;
  if t(j+1)>t(j)
    if j==lenge-Ordnung %am rechten Rand linksseitig stetig!
      J=n(j):n(j+1);
      else
      J=n(j):n(j+1)-1;
    end; %if
    B(j,J)=ones(1,length(J));
  end; %if t(...
end; %for j
%==========================================================================
%Globalschleife f\"ur alle Ordnungen
for k=2:Ordnung
  %========================================================================
  %Bestimmung der Omegas
  for j=1:lenge-k+1
    if t(j+k-1)>t(j)
      omega(j,:)=(tt-t(j))/(t(j+k-1)-t(j));
      else
      omega(j,:)=zeros(1,l);
    end; %if t(
  end; %for j
%==========================================================================
%Bestimmen der Splines: Rekursionsformel
  for j=1:lenge-k
    B(j,:)=omega(j,:).*B(j,:)+(1-omega(j+1,:)).*B(j+1,:);
  end; %for j
%==========================================================================
end; %for k
%==========================================================================
toc;  %Die Stoppuhr wird gestoppt.
%==========================================================================
%Globalschleife f\"ur alle Ordnungen zu Ende
%Alle B-Slines sind berechnet, sie sind in einer Matrix B(j,l) zusammengefa\ss t.
```

```
 79 %j=1,2,...,lenge-ordnung; lenge=Anzahl der Knoten, inkl. der zus\"atzlichen
 80 %Randknoten, l=100*(b-a)+1, dabei werden die Randknoten a, b und alle
 81 %anderen Knoten als ganzzahlig vorausgesetzt.
 82 %==========================================================================
 83 %Zeichnen der Splines B(j,:), j=1,2,...,lenge-Ordnung
 84 %==========================================================================
 85 %Zuerst Knoten zeichnen
 86 figure(1)
 87 for j=1:lenge
 88   plot([t(j),t(j)],[0,0.5],'r');  %Trennstriche zwischen den Knoten
 89                                   %Die Zahl 0.5 mu\ss\ evtl angepa\ss t werden
 90   if j==1,hold on, end; %if
 91 end; %for j
 92 plot(t,zeros(size(t)),'ro');
 93 %==========================================================================
 94 %Alle (oder einige) Splines der Ordnung k in eine Zeichnung
 95 for j=6%:lenge-Ordnung
 96   plot(tt,B(j,:),'k'); %'k' hei\ss t "schwarz"
 97 end; %for j
 98 %==========================================================================
 99 %Ende der Zeichnung
100 %==========================================================================
101 %Will man die Polynomst\"ucke zwischen benachbarten Knoten wiederfinden, so ist
102 %das am einfachsten, wenn man ein Interpolationsproblem mit polyfit l\"ost.
103 %Beispiel f\"ur den Fall der Aufgabe 17.
104 Stellen=[1,10,20,40,80,160]; %6 beliebige Werte zwischen Null und zwei
105 %Diese Stellen entsprechen den Argumenten [1,10,20,40,80,160]/100.
106 u=tt(Stellen);v=B(6,Stellen);
107 p=polyfit(u,v,5);
108 z=linspace(0,6.2);
109 w=polyval(p,z);
110 plot(z,w,'--m');   %gestrichelt mit Farbe "magenta"
111 text(6,0.48,'#1'); %das mu\ss\ ausprobiert werden
112 hold off;
```

Die angegebenen Rechnungen (alle B-Splines bis zur Ordnung 6 an insgesamt 1601 Stellen, ohne Zeichnungen) dauern mit MATLAB, Version 5.3 auf einer Sun ULTRA1, unix-Betriebssystem Solaris 2.51, Risc-Prozessor, 140 MHz, 0.26 Sekunden. Wie man die Polynomstücke zeichnen kann, steht am Ende im obigen Programm L9.3 ab Zeile 101.

Aufgabe 18. Bestimmen Sie eine B-Spline-Darstellung der Polynome (a) $p(t) := 3t^2 - 2t + 1$, (b) $p(t) := 5t^3 - 3t + 2$. Wählen Sie (A) ganzzahlige und (B) $\pm$ primzahlige Knoten also $\mathbf{t} := \{\ldots, -7, -5, -3, -2, 2, 3, 5, 7, \ldots\}$.

Lösung. Die Formeln zur Berechnung der gesuchten Darstellungen haben die Nummern (9.3) und (9.4). Da $\lambda_{jk}[p]$ nicht von τ abhängt (man leite den Ausdruck für $\lambda_{jk}[p]$ nach τ ab), muß man nur den konstanten Teil berechnen. Die Darstellung von p in $[t_0, t_1]$ lautet dann: $p = \sum_{\ell=-k+1}^{0} \lambda_{\ell k}[p] B_{\ell k}$.

Fall (a), Ordnung $k = 3$:

$$\begin{aligned}
\psi_{j3}(t) &= (t_{j+1} - t)(t_{j+2} - t), \quad \psi'_{j3}(t) = 2t - (t_{j+1} + t_{j+2}), \quad \psi''_{j3}(t) = 2,\\
p(t) &= 3t^2 - 2t + 1, \quad p'(t) = 6t - 2, \quad p''(t) = 6,\\
\lambda_{j3}[p] &= (1/2)(p''(t)\psi_{j3}(t) - p'(t)\psi'_{j3}(t) + p(t)\psi''_{j3}(t)) = 3t_{j+1}t_{j+2} - (t_{j+1} + t_{j+2}) + 1\\
&= (\text{Fall (A)}) : \lambda_{:,3} = [2, 0, 4], (\text{Fall (B)}, t_0 = -2) : \lambda_{:,3} = [24, -11, 14].
\end{aligned}$$

Fall (b), Ordnung $k = 4$:

$$\begin{aligned}
\psi_{j4}(t) &= (t_{j+1} - t)(t_{j+2} - t)(t_{j+3} - t), \quad \psi'_{j4}(t) = (t_{j+1} - t)(2t - (t_{j+2} + t_{j+3})) - (t_{j+2} - t)(t_{j+3} - t),\\
\psi''_{j4}(t) &= -6t + 2(t_{j+1} + t_{j+2} + t_{j+3}), \quad \psi'''_{j4}(t) = -6,\\
p(t) &= 5t^3 - 3t + 2, \quad p'(t) = 15t^2 - 3, \quad p''(t) = 30t, \quad p'''(t) = 30,\\
\lambda_{j4}[p] &= (1/6)(p'''(t)\psi_{j4} - p''(t)\psi'_{j4}(t) + p'(t)\psi''_{j4}(t) - p(t)\psi'''_{j4}(t))\\
&= 5t_{j+1}t_{j+2}t_{j+3} - (t_{j+1} + t_{j+2} + t_{j+3}) + 2\\
&= (\text{Fall (A)}) : \lambda_{:,4} = [5, 2, -1, 26], (\text{Fall (B)}, t_0 = -2) : \lambda_{:,4} = [-138, 65, -61, 142].
\end{aligned}$$