Numerische Mathematik für StudentInnen mit Ziel Diplom in Mathematik

Gerhard Opfer — WS 1997/1998 (1. Teil) — SS 1998 (2. Teil)

Bogen 5: Übungen am 9. Dezember 1997

Lösungshinweise

Aufgabe P13. Die beiden Formen lauten:

$$\begin{aligned}
p(x) = {} & 2\frac{(x+1)(x-3)(x-4)}{(-2+1)(-2-3)(-2-4)} + \frac{(x+2)(x-3)(x-4)}{(-1+2)(-1-3)(-1-4)} + \\
& 3\frac{(x+2)(x-1)(x-4)}{(3+2)(3-1)(3-4)} + 5\frac{(x+2)(x+1)(x-3)}{(4+2)(4+1)(4-3)} \\
= {} & \frac{-(x+1)(x-3)(x-4)}{15} + \frac{(x+2)(x-3)(x-4)}{20} + \\
& -3\frac{(x+2)(x-1)(x-4)}{10} + \frac{(x+2)(x+1)(x-3)}{6}, \\
p(x) = {} & 2-(x+2)+0.3(x+2)(x+1).
\end{aligned}$$

Aufgabe P14. Man löst das Interpolationsproblem mit den Daten $(2,-2)$, $(1,-1)$, $(3,3)$, $(5,4)$. Die Newtonform lautet $p(y) = -2-(y-2)+3(y-2)(y-1)-(27/24)(y-2)(y-1)(y-3)$. Daraus folgt $x = p(4) = 7.25$, ein Wert außerhalb des betrachteten Intervalls. An den Stellen $1, 2, 3, 4$ hat p die Werte $0.8, 1.6, 3, 5$. Machen wir dieselbe Rechnung mit diesen Daten, so erhält man $x \approx 3.4600$ und $p(x) = 3.8453$, eine immerhin grobe Näherung.

Aufgabe P15. Das Neville-Schema sieht hier für $p(0) = 0.6$ so aus:

-2	2	0	0.6	0.6
-1	1	1.5	0.6	
3	3	-3		
4	5			

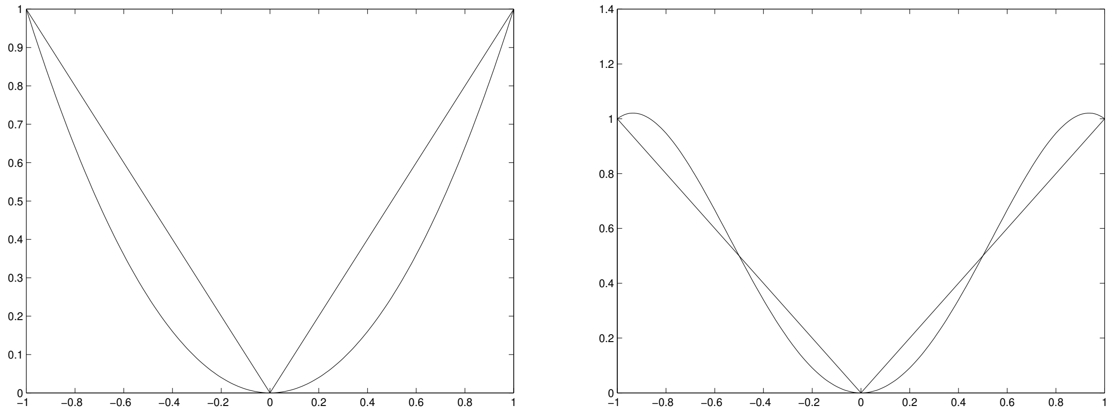
Aufgabe 9. Das Programm und die Bilder für $n = 2, 4, 8, 16$ sind unten angehängt.

Aufgabe 10. Die angegebene Summe interpoliert offenbar die Werte $y_0, y_1, \ldots, y_n = 1, 1, \ldots, 1$, daher muß die Summe mit der konstanten Funktion Eins übereinstimmen. Der zweite Teil ist eine einfache Anwendung der Dreiecksungleichung.

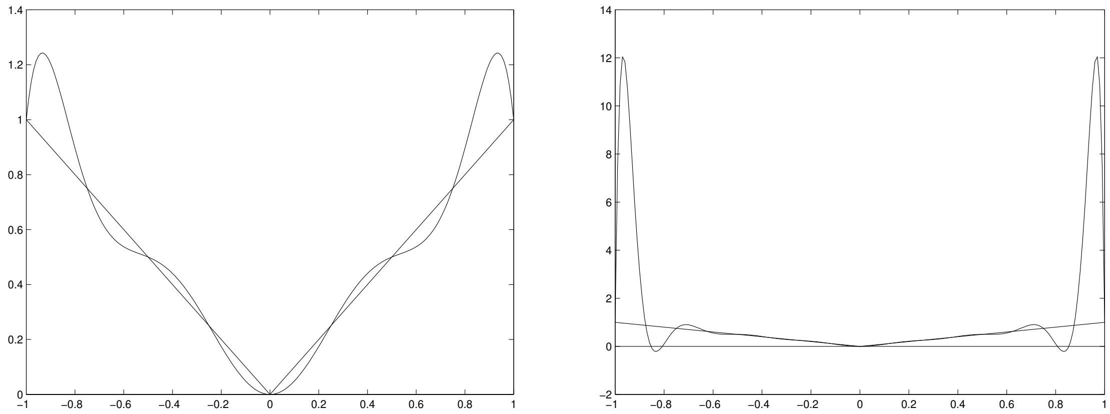
```
program Newton_Interpolation;
{Numerik WS 1997/1998, Aufgabe 9}
{Zuerst werden die dividierten Differenzen ausgerechnet, (div_diff)}
{dann wird das Newton-Polynom ausgewertet (newton_pol), zum Schlu werden}
{Daten zur Testrechnung bereitgestellt (Test_Daten). Zum Vergleich wird}
{auch der Neville-Algorithmus, rekursiv programmiert.}
const nmax=32;dis=200;
type vektor=array[0..nmax] of real;
     vektordis=array [0..dis] of real;
var kn, we, d:vektor;x,y:vektordis;
    xx,yy:real;
    n:integer;
{-------------------------------------------------------------------}
procedure div_diff(n:integer;Knoten,Werte:vektor; var d_diff:vektor);
{Es werden die dividierten Differenzen, die man zur Newton-Interpolation}
{bentigt, ausgerechnet ohne Beanspruchung von zustzlichen Speicherpltzen}
```

```
  var j,k:integer;
  begin
    d_diff:=Werte;
    for j:=1 to n do
      for k:=n downto j do
        d_diff[k]:=(d_diff[k]-d_diff[k-1])/(Knoten[k]-Knoten[k-j]);
  end; {procedure}
{----------------------------------------------------------------------}
function newton_pol(n:integer;x:real;Knoten,d_diff:vektor):real;
{Die Auswertung des Newton-Polynoms p wird mit einem Horner-hnlichen}
{Algorithmus vorgenommen:}
{p(x) = c0 +(x-x0)(c1+(x-x1)(c2+...+(x-x[n-2])(c[n-1]+cn(x-x[n-1]))...)}
var j:integer;S:real;
begin
  S:=d_diff[n];
  for j:=n-1 downto 0 do
    S:=d_diff[j]+(x-Knoten[j])*S;
  newton_pol:=S;
end; {function}
{----------------------------------------------------------------------}
function neville(j,k:integer;x:real;Knoten,Werte:vektor):real;
{Der Wert p(x) des Interpolationspolynoms vom Grad k durch die Knoten}
{x_j,...x_j+k wird mit Hilfe der Nevilleschen Formel berechnet.}
{Ein direkter, nicht rekursiver Algorithmus steht im Buch, 2. Aufl., S. 40}
begin
  if k=0 then neville:=Werte[j] else
  neville:=((x-Knoten[j])*neville(j+1,k-1,x,Knoten,Werte)
  -(x-Knoten[j+k])*neville(j,k-1,x,Knoten,Werte))/(Knoten[j+k]-Knoten[j]);
end; {function}
{----------------------------------------------------------------------}
procedure x_und_y_Werte_ausrechnen(n,dis:integer;Knoten,d_diff:vektor;
                                   var x,y: vektordis);
{Es werden fr alle Werte von x[j], j=0,1,...,dis die Werte y[j] = p(x[j])}
{berechnet und hintereinander in l5.erg abgespeichert.}
{Dient zur Vorbereitung zum Zeichnen.}
var j:integer;
    h:real;
    dat:text;
begin
  rewrite(dat,'l5.erg');
  h:=2/dis;
  for j:=0 to dis do
  begin
    x[j]:=-1+j*h;
    y[j]:=newton_pol(n,x[j],Knoten,d_diff);
  end;
  for j:=0 to dis do
    writeln(dat,x[j]); {abspeichern in Datei dat=l5.erg}
  for j:=0 to dis do
    writeln(dat,y[j]);
  close(dat);
  {%Das Herstellen eines Bildes geht mit MATLAB sehr einfach:}
  {load l5.erg %erzeugt einen Spaltenvektor l5 der x und y}
  {%(hintereinander) enthlt.}
  {dis=(length(l5)-2)/2;}
  {x=l5(1:dis+1);y=l5(dis+2:2*dis+2);}
  {plot(x,y,[-1 0],[1 0],[0 1],[0 1],[-1 1],[0 0]);}
  {%Es wird auch der Graph von abs(x) und die x-Achse gezeichnet.}
  {%Mit print kann direkt gedrucktwerden, mit}
  {print -deps bild.ps}
  {%wird eine druckbare Datei 'bild.ps' hergestellt.}
  {%Lt man die geschweiften Klammern weg,}
  {%so erhlt man ein ausfhrbares MATLAB-Programm}
end; {procedure}
{----------------------------------------------------------------------}
procedure Test_Daten(Fall:boolean;var n:integer;var Knoten,Werte:vektor);
{Durch die Variable Fall knnen verschieden Beispieltypen behandelt werden}
var j:integer;
    h:real;
begin
if Fall then
  begin
    n:=3;Knoten[0]:=0.8;Werte[0]:= 1;
```

```
         Knoten[1]:=1.6;Werte[1]:= 2;
         Knoten[2]:=  3;Werte[2]:= 3;
         Knoten[3]:=  5;Werte[3]:= 4; {Aufgabe P14}
    n:=3;Knoten[0]:=-2; Werte[0]:= 2;
         Knoten[1]:=-1; Werte[1]:= 1;
         Knoten[2]:= 3; Werte[2]:= 3;
         Knoten[3]:= 4; Werte[3]:= 5; {Aufgabe P13}
  end else
  begin
    n:=16; h:=2/n;
    for j:=0 to n do
    begin
      Knoten[j]:=-1+j*h; Werte[j]:=abs(Knoten[j]);
    end;
  end;
end; {Test_Daten}
{---------------------------------------------------------------------}
begin {Hauptprogramm}
  Test_Daten(false,n,kn,we); {Die Variablen n, kn, we werden besetzt}
  div_diff(n,kn,we,d); {Die dividierten Differenzen d werden berechnet}
  x_und_y_Werte_ausrechnen(n,dis,kn,d,x,y); {Das Interpolationspolynom}
  {wird an dis + 1 quidistanten Stellen ausgewertet}
  xx:=-1+1/n;           {zwischen den ersten beiden Knoten}
                        {wird ein Probewert berechnet.}
  yy:=newton_pol(n,xx,kn,d);
  writeln(yy,' ',abs(xx)-yy,'     Testwert und Fehler');
  yy:=neville(0,n,xx,kn,we);
  writeln(yy,                ''     Test Neville');
end.
```



Interpolation der abs-Funktion mit $n = 2$ und $n = 4$



Interpolation der abs-Funktion mit $n = 8$ und $n = 16$