**Numerische Mathematik für StudentInnen**
**mit Ziel Diplom in Mathematik**

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

---

**Bogen 6: Übungen am 16. Dezember 1997**

**Lösungshinweise**

---

**Aufgabe P16.** Es ist bei ungeradem $n$

$$
\begin{aligned}
|\omega(0.5)| &= \frac{1}{2}\frac{1}{2}\frac{3}{2}\frac{5}{2}\cdots\frac{2n-1}{2} = \frac{1\cdot 3\cdots 2n-1}{2^{n+1}} = \frac{(2n)!}{2^{2n+1}n!},\\
|\omega(n/2)| &= \frac{n}{2}\frac{n-2}{2}\cdots\frac{1}{2}\frac{1}{2}\cdots\frac{n-2}{2}\frac{n}{2} = \Big(\frac{n!}{2^n(\frac{n-1}{2})!}\Big)^2.
\end{aligned}
$$

Der Quotient ist entsprechend

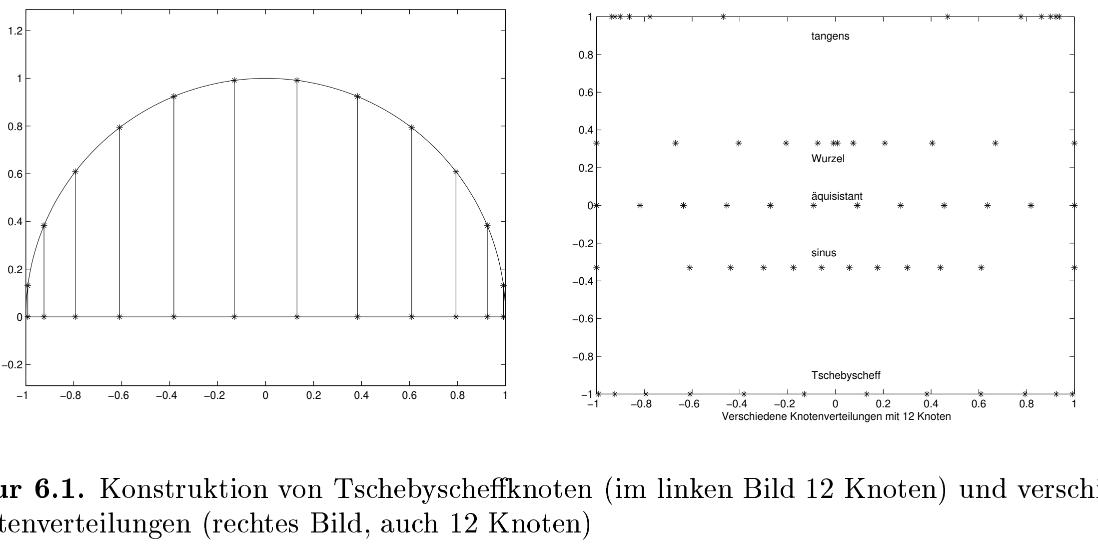
$$
\frac{|\omega(0.5)|}{|\omega(n/2)|} = \frac{\dfrac{(2n)!}{2^{2n+1}n!}}{\Big(\dfrac{n!}{2^n(\frac{n-1}{2})!}\Big)^2} = \frac{(2n)!((\frac{n-1}{2})!)^2}{2(n!)^3}.
$$

Für $n=3$ erhält man für diese drei Zahlen z. B. $6!/(2^7 3!) = 15/16$, $(3!/8)^2 = 9/16$, $6!(1!)^2/(2(3!)^3) = 5/3$. Für $n=7$ sind die entsprechenden Zahlen 527.87109375, 43.06640625, 12.257142857143.

**Aufgabe P17.** Mit der Neville-Formel durch Vergleich der Höchstkoeffizienten.

**Aufgabe P18.** Im Lösungsbogen 5 enthhalten (Aufgabe 10).

**Aufgabe 11.** Da $z_j$ auf dem Kreisrand liegt ist $y_j = \sin\arccos x_j$. Teilt man den gesamten Kreis äquidistant in $2n+2$ Winkel, so daß die $x$-Achse einen dieser Winkel symmetrisch zerlegt, so sind die Projektionen von den Teilungspunkten der Kreislinie auf die $x$-Achse die gesuchten Knoten. Man vergleiche die Figur 6.1, linkes Bild.




**Figur 6.1.** Konstruktion von Tschebyscheffknoten (im linken Bild 12 Knoten) und verschiedenen Knotenverteilungen (rechtes Bild, auch 12 Knoten)

**Aufgabe 12.** Die vorgeschlagenen Knotenwahlen sehen aus wie in Figur 6.1, rechts. Ein Programm zur Bestimmung der Maxima bei diversen Knotenwahlen ist angefügt. Für fünf Knotenwahlen sind die Rechenergebnisse an das Programm angehängt.

**Programm L6.1** Berechnung des Maximums eines Knotenpolynoms

```
{Auswertung eines Knotenpolynoms und Bestimmung des Maximums.}
{Numerik WS97/98, Aufgabenbogen 6}
program Knotenpolynom;
const nmax=32; dis=10000;
type vektor=array[0..nmax] of real;
     ivektor=array[1..6] of integer;
     vektordis=array[0..dis] of real;
var  j,n,Fall,jmax:integer;
     Kno:vektor;werte:vektordis;
     jj:ivektor;
     wmax:real;
     dat:text;

function sign(x:real):real;
begin
  if x>0 then sign:=1;
  if x=0 then sign:=0;
  if x<0 then sign:=-1;
{Ginge auch mit abs(x)/x fr x not= 0, braucht aber mehr Rechenzeit}
end; {function}

function arcsin(y:real):real;
{Hilfskonstruktion, Abramowitz & Stegun, S. 81, Fehler in [-1,1] <= 2e-8.}
const a0= 1.5707963050;
      a1=-0.2145988016;
      a2= 0.0889789874;
      a3=-0.0501743046;
      a4= 0.0308918810;
      a5=-0.0170881256;
      a6= 0.0066700901;
      a7=-0.0012624911;
var p,x,pihalbe:real;
begin
  pihalbe:=2*arctan(1);
  x:=abs(y);
  if x>1 then
  writeln('arcsin nicht definiert fr x = ',y) else
  begin
    p:=a0+x*(a1+x*(a2+x*(a3+x*(a4+x*(a5+x*(a6+a7*x))))));
    p:=pihalbe-sqrt(1-x)*p;
    if y<0 then arcsin:=-p else arcsin:=p;
  end; {else}
end; {function}

procedure Knoten_Besetzung(n,Fall:integer;var Knoten:vektor);
{Die Knoten werden besetzt}
var j:integer;
    pi,h:real;
begin
  pi:=4*arctan(1);
  for j:=0 to n do
  begin
    h:=-1+2*j/n;
    case Fall of
      1:  Knoten[j]:=cos((2*(n-j)+1)/(2*n+2)*pi);          {Tschebyscheff}
      2:  Knoten[j]:=h;                                {quidistante Knoten}
      3:  Knoten[j]:=arcsin(h);{arcsin in Standard-Pascal nicht definiert}
      4:  Knoten[j]:=sign(h)*sqr(h);
      5:  Knoten[j]:=arctan(10*h)/pi*2;
      otherwise writeln('Knoten nicht definiert!');
      {else writeln('Knoten nicht definiert!');}
    end; {case}
  end; {for j}
end; {procedure}

procedure Werte_Knotenpolynom(n:integer;Knoten:vektor;var erg:vektordis);
{Es werden dis+1 Werte des Betrages des Knotenpolynoms an quidistanten Stellen}
```

```
{in [-1,1] ausgerechnet und auf erg abgelagert}
{----------------------------------------------------------------------}
function kn(n:integer;x:real;Knoten:vektor):real;
{Mit den Knoten Knoten[0],...,Knoten[n] soll}
{omega(x)=(x-Knoten[0])(x-Knoten[1])...(x-Knoten[n]) ausgewertet werden.}
var p:real;
    j:integer;
begin
  p:=1;
  for j:=0 to n do
    p:=p*(x-Knoten[j]);
  kn:=p;
end; {function kn}
{----------------------------------------------------------------------}
var j:integer;
begin
  for j:=0 to dis do
    erg[j]:=abs(kn(n,-1+2*j/dis,Knoten));
end; {procedure Werte_Knotenpo}

procedure maximum(v:vektordis;var jj:integer; var m:real);
{Es wird die maximale Komponente von v bestimmt. m=v(jj)=max(v)}
{Gibt es mehrere Maxima, so wird das erste genommen}
var j,jmax:integer;
    max,h: real;
begin
  max:=0; {funktioniert nicht, wenn das Maximum negativ ist}
  for j:=0 to dis do
  begin
    h:=v[j];
    if h>max then
    begin max:=h;jmax:=j;end;
  end; {for j}
jj:=jmax;m:=max;
end; {procedure maximum}

begin {Hauptprogramm}
  rewrite(dat,'l6.erg');
  jj[1]:=2;jj[2]:=3;jj[3]:=8;jj[4]:=9;jj[5]:=16;jj[6]:=17;
  for Fall:=1 to 5 do {die verschiedenen Knotensorten, Fall 1=Tschebyscheff}
  begin
    for j:=1 to 6 do
    begin
      n:=jj[j];   {n+1 = Grad des Knotenpolynoms}
      Knoten_Besetzung(n,Fall,Kno);
      Werte_Knotenpolynom(n,Kno,werte);
      maximum(werte,jmax,wmax);
      writeln(    'Fall = ',Fall:3,' Grad n+1 = ,',n+1:3,' Maximum = ',wmax:8:5,
             ' bei x = ',-1+2*jmax/dis:8:5);
      writeln(dat,'Fall = ',Fall:3,' Grad n+1 = ,',n+1:3,' Maximum = ',wmax:8:5,
             ' bei x = ',-1+2*jmax/dis:8:5);
    end; {for j}
  writeln;writeln(dat);
  end; {for Fall}
  close(dat);
end.
{Rechenergebnisse:
Fall =   1 Grad n+1 = ,  3 Maximum =  0.25000 bei x = -0.50000 T-Knoten
Fall =   1 Grad n+1 = ,  4 Maximum =  0.12500 bei x = -1.00000
Fall =   1 Grad n+1 = ,  9 Maximum =  0.00391 bei x = -1.00000
Fall =   1 Grad n+1 = , 10 Maximum =  0.00195 bei x = -1.00000
Fall =   1 Grad n+1 = , 17 Maximum =  0.00002 bei x = -1.00000
Fall =   1 Grad n+1 = , 18 Maximum =  0.00001 bei x =  0.00000

Fall =   2 Grad n+1 = ,  3 Maximum =  0.38490 bei x =  0.57740 quidistante Knoten
Fall =   2 Grad n+1 = ,  4 Maximum =  0.19753 bei x =  0.74540
Fall =   2 Grad n+1 = ,  9 Maximum =  0.01880 bei x =  0.92480
Fall =   2 Grad n+1 = , 10 Maximum =  0.01260 bei x =  0.93500
Fall =   2 Grad n+1 = , 17 Maximum =  0.00094 bei x = -0.96780
Fall =   2 Grad n+1 = , 18 Maximum =  0.00066 bei x =  0.97020

Fall =   3 Grad n+1 = ,  3 Maximum =  1.49179 bei x = -0.90680 Sinus-Knoten
Fall =   3 Grad n+1 = ,  4 Maximum =  1.29793 bei x = -1.00000
```

```
Fall =   3 Grad n+1 = ,  9 Maximum =  0.27998 bei x = -1.00000
Fall =   3 Grad n+1 = , 10 Maximum =  0.17236 bei x =  1.00000
Fall =   3 Grad n+1 = , 17 Maximum =  0.01727 bei x = -1.00000
Fall =   3 Grad n+1 = , 18 Maximum =  0.01497 bei x =  1.00000

Fall =   4 Grad n+1 = ,  3 Maximum =  0.38490 bei x =  0.57740 Wurzelknoten
Fall =   4 Grad n+1 = ,  4 Maximum =  0.24387 bei x =  0.71140
Fall =   4 Grad n+1 = ,  9 Maximum =  0.05086 bei x = -0.89940
Fall =   4 Grad n+1 = , 10 Maximum =  0.03933 bei x =  0.91180
Fall =   4 Grad n+1 = , 17 Maximum =  0.00777 bei x = -0.95400
Fall =   4 Grad n+1 = , 18 Maximum =  0.00626 bei x =  0.95720

Fall =   5 Grad n+1 = ,  3 Maximum =  0.31618 bei x = -0.54080 Tangens-Knoten
Fall =   5 Grad n+1 = ,  4 Maximum =  0.58183 bei x =  0.00000
Fall =   5 Grad n+1 = ,  9 Maximum =  0.05785 bei x = -0.28680
Fall =   5 Grad n+1 = , 10 Maximum =  0.10984 bei x =  0.00000
Fall =   5 Grad n+1 = , 17 Maximum =  0.00611 bei x = -0.19460
Fall =   5 Grad n+1 = , 18 Maximum =  0.00937 bei x =  0.00000
arcsin(j/10), j=0,1,,,,10:
 x      exakt(matlab)       Approximation (mit diesem Programm)
0.0:  0                     0.00000002
0.1:  0.10016742116156      0.10016743
0.2:  0.20135792079033      0.20135792
0.3:  0.30469265401540      0.30469264
0.4:  0.41151684606749      0.41151685
0.5:  0.52359877559830      0.52359879
0.6:  0.64350110879328      0.64350110
0.7:  0.77539749661075      0.77539749
0.8:  0.92729521800161      0.92729522
0.9:  1.11976951499863      1.11976952
1.0:  1.57079632679490 (pi/2)
}
```