Numerische Mathematik für StudentInnen
mit Ziel Diplom in Mathematik

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

Bogen 2: Übungen am 11. und 18. November 1997

Lösungshinweise

Aufgabe P4. Wegen $P(x) = (x - \xi_1)(x - \xi_2) = x^2 - (\xi_1 + \xi_2)x + \xi_1\xi_2$ gilt $\xi_1\xi_2 = q$, und für $q \neq 0$ folgt $\xi_2 = q/\xi_1$ (eine der Formeln von Vieta, die andere ist $\xi_1 + \xi_2 = 2p$). Setzt man $h = \sqrt{p^2 - q}$, so rechnet man $\xi_1 = p + \text{sign}(p)h$; $\xi_2 = q/\xi_1$. Für die angegebenen Größen erhält man (leicht editiert, mit Sun-Pascal)

$p := \ \ 1 : \xi_1 = \ \ 1.999999999950, \xi_2 = \ \ 5.00000000012500e{-11}$, direkt: $\ \ 5.00000041370185e{-11}$,

$p := -1 : \xi_1 = -1.999999999950, \xi_2 = -5.00000000012500e{-11}$, direkt: $-5.00000041370185e{-11}$.

Aufgabe P5.

2	-3	0	1	-4
	6	9	27	84
2	3	9	28	$\boxed{p(3) = 80}$
	6	27	105	
2	9	35	$\boxed{p'(3) = 133}$	
	6	45		
2	15	$\boxed{p''(3)/2! = 80}$		
	6			
2	$\boxed{p'''(3)/3! = 21}$			

$\boxed{p^{iv}(3)/4! = 2}$

Aufgabe P6. Man lese das Buch, ggf auch den Abschnitt über komplexe Zahlen.

Aufgabe 3.

```
program Reduktion;
{Pascalprogramm zum Reduzieren einer Gleitkommazahl auf n>=1 Stellen}
{Gerhard Opfer: Numerik 1, WS 1997/1998, Aufgabe 3}
{Zuerst einige Hilfsprogramme}
{--------------------------------------------------------------------------------}
{1. 10^n} function zehnhochn(n:integer):real;
var
  h,c:real; j:integer;
begin
  h:=1; c:=10; if n<0 then begin c:=1/10; n:=-n;end;
  for j:=1 to n do
    h:=h*c;
  zehnhochn:=h;
end; {function}
{--------------------------------------------------------------------------------}
{2. Zehnerlogarithmus log} function log(x:real):real;
const M=0.4342944819032518276511289;  {M=log_[10](e), A. & St,. S. 3}
begin
  log:=M*ln(x);
end; {function}
{--------------------------------------------------------------------------------}
```

```
{3. Exponent weg: x=z.z...zEn ---> yo=z.z...z}
procedure exponentweg(x:real;var n:integer;var yo:real);
var h,hh:real;ex:integer;
begin
  h:=abs(x);
  hh:=log(h);
  ex:=trunc(hh);
  if (hh<0) and (hh <> ex) then  {<> bedeutet ungleich}
    ex:=ex-1;
  yo:=x*zehnhochn(-ex);n:=ex;
end; {procedure}
{----------------------------------------------------------------------------}
{4. Exponent wieder dran: yo=z.z...z ---> x=z.z...zEn}
function exponentdran(yo:real;n:integer):real;
var h:real;
begin
  h:=abs(yo);
  if (h<10) and (h>=1) then
    exponentdran:=yo*zehnhochn(n);
end; {function}
{----------------------------------------------------------------------------}
{5. round: Die Standardprozedur round liefert ganzzahlige Werte und ist daher  }
    {daher nur im ganzzahligen Bereich von Pascal (richtig) definiert. Fr zu  }
    {groe Argumente liefert round falsche Werte und ggf. eine Warnung.        }
    {Fr groe n (>10) mte eine eigene round-Prozedur geschrieben werden.    }
{----------------------------------------------------------------------------}
{6. eigentliches Programm}
{Die Zahl x soll (dezimal) auf n Mantissenstellen reduziert werden}
function reduktion(x:real;n:integer):real;
var h1,h2,yo:real;nh:integer;
begin
  if n>=10 then
              writeln('Ergebnis wg. falscher Rundung voraussichtlich falsch!');
  if n<=0 then writeln('Sinnloser Aufruf, Stellenzahl = ',n);
  if x=0 then reduktion:=0 else
    begin
      exponentweg(x,nh,yo);
      h1:=zehnhochn(n-1); h2:=round(h1*yo)/h1;
      reduktion:=exponentdran(h2,nh);
    end; {if}
end; {function}
{----------------------------------------------------------------------------}
  var
    testrein,testraus:real;ntest:integer;
begin {des Hauptprogramms}
  {Wir probieren einige Testflle}
  testrein:=10000*4*arctan(1); {=pi*10000}
  writeln(' ');
  for ntest:=0 to 5 do
  begin
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
  end; {for}
  testrein:=(1/10000)*4*arctan(1); {=pi/10000}
  writeln(' ');
  for ntest:=1 to 5 do
  begin
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
  end; {for}
  testrein:=-10000*4*arctan(1); {=-pi*10000}
  writeln(' ');
  for ntest:=1 to 5 do
  begin
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
  end; {for}
  testrein:=-(1/10000)*4*arctan(1); {=-pi/10000}
  writeln(' ');
  for ntest:=1 to 5 do
  begin
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
```

```
  end; {for}
  writeln(' ');
  for ntest:=1 to 5 do
  begin
    testrein:=ntest;
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
  end; {for}
  writeln(' ');
  for ntest:=1 to 5 do
  begin
    testrein:=-zehnhochn(-ntest);
    testraus:=reduktion(testrein,2*ntest);
    writeln(2*ntest,' ',testrein,' ',testraus);
  end; {for}
end. {des Hauptprogramms}
{---------------------------------------------------------------------------------}

{Testergebnisse:
         n          x                  reduktion(x)
-------------------------------------------------------
Sinnloser Aufruf, Stellenzahl =          0
         0  3.14159265358979e+04  1.00000000000000e-01
         2  3.14159265358979e+04  3.10000000000000e+04
         4  3.14159265358979e+04  3.14200000000000e+04
         6  3.14159265358979e+04  3.14159000000000e+04
         8  3.14159265358979e+04  3.14159270000000e+04
Ergebnis wg. falscher Rundung voraussichtlich falsch!
        10  3.14159265358979e+04  2.14748364700000e+04

         2 -3.14159265358979e-04 -3.10000000000000e-04
         4 -3.14159265358979e-04 -3.14200000000000e-04
         6 -3.14159265358979e-04 -3.14159000000000e-04
         8 -3.14159265358979e-04 -3.14159270000000e-04
Ergebnis wg. falscher Rundung voraussichtlich falsch!
        10 -3.14159265358979e-04 -2.14748364800000e-04

         2  1.00000000000000e+00  1.00000000000000e+00
         4  2.00000000000000e+00  2.00000000000000e+00
         6  3.00000000000000e+00  3.00000000000000e+00
         8  4.00000000000000e+00  4.00000000000000e+00
Ergebnis wg. falscher Rundung voraussichtlich falsch!
        10  5.00000000000000e+00  2.14748364700000e+00

         2 -1.00000000000000e-01 -1.00000000000000e-01
         4 -1.00000000000000e-02 -1.00000000000000e-02
         6 -1.00000000000000e-03 -1.00000000000000e-03
         8 -1.00000000000000e-04 -1.00000000000000e-04
Ergebnis wg. falscher Rundung voraussichtlich falsch!
        10 -1.00000000000000e-05 -1.00000000000000e-05

Fehlermeldung aus Sun-Pascal:
Note: the following IEEE floating-point arithmetic exceptions
occurred and were never cleared; see ieee_flags(3M):
Inexact;  Invalid Operand;
Sun's implementation of IEEE arithmetic is discussed in
the Numerical Computation Guide.
}
```

Aufgabe 4. (a) Für $n = 100$ erhält man keine Fehler, für $n = 200, 300$ sind die Fehler minimal:

$$\delta_{200} = -1.11022302462516e - 16, rel^1 = 0, rel^2 = -1.11577413974828e - 16,$$

$$\delta_{300} = -1.11022302462516e - 16, rel^1 = 0, rel^2 = -1.11392376804057e - 16.$$

(b) Nach 62 Schritten mit $x = -15$: $3.0590\,94704\,08486e - 07$, Nach 62 Schritten mit $x = 15$: $3.0590\,23205\,01826e - 07$. Die zweite Summe ist vollständig (in der angegebenen Stellenzahl) richtig, die erste hat nur vier richtige Stellen.