

## Das Horner-Schema

horner.py, Horner1.py

Mit dem Horner-Schema kann man ein Polynom in seine (Linear-) Faktoren zerlegen. Natürlich geht das mit einem CAS-Rechner direkt, aber es macht Spaß, den Algorithmus in einem eigenen Programm umzusetzen.

$$\begin{array}{r|l}
 a & \dots \dots \dots \dots \dots \dots \leftarrow P(x) \\
 & \dots \dots \dots \dots \dots \dots \leftarrow +a \cdot Q(x) \\
 \hline
 Q(x) \rightarrow & \dots \dots \dots \dots \dots \dots \leftarrow R
 \end{array}$$

Wenn  $R = 0$ , dann ist das Polynom durch  $(x - a)$  teilbar und man gelangt zur Faktorisierung  $P(x) = (x - a) \cdot Q(x)$ . Die wiederholte Anwendung dieses Verfahrens ergibt die vollständige Zerlegung – wenn das Polynom nur ganze Nullstellen hat.

```

28.2 29.1 29.2 *PyKurz RAD
horner.py erfolgreich gespeichert
while 1:
  k=input("Koeffizienten: "); p=eval("[ "+k+" ]")
  a=-abs(p[-1])
  while a<=abs(p[-1]):
    q=p[:]
    for i in range(1,len(q)):
      q[i]=p[i]+a*q[i-1]
    if q[-1]=0:a+=1
    else:print(1,-a);p=q[:-1];a=-abs(p[-1])
  print(p);print("")
    
```

```

28.2 29.1 29.2 *PyKurz RAD 13/13
Python-Shell
Koeffizienten: 1,1,-4,-4
[1, 2]
[1, 1]
[1, -2]
[1]
>>>
Koeffizienten: 1,3,2,6
[1, 3]
[1, 0, 2]
>>>
Koeffizienten: |
    
```

Nur die Linearfaktoren mit ganzzahligen Nullstellen werden angezeigt:

$$\begin{aligned}
 x^3 + x^2 - 4x - 4 &= (x + 2) \cdot (x + 1) \cdot (x - 2) \\
 x^3 + 3x^2 + 2x + 6 &= (x + 3) \cdot (x^2 + 2)
 \end{aligned}$$

- Wir beginnen mit einer offenen while-Schleife (ohne festes Ende). Dann wird der Benutzer zur Eingabe der Koeffizienten des Polynoms aufgefordert: 1,1,-4,-4 steht daher für das Polynom  $x^3 + x^2 - x - 4$ . Mit eval wird die Eingabe in die Python-Liste  $p$  umgewandelt. (Beachte den Hinweis am Ende des Beispiels.)
- $p[-1]$  ist das letzte Element der Liste und damit der konstante Term  $c$  des Polynoms. Wir beginnen mit dem Kandidat Hornerzahl  $a = |c|$ , weil wir uns nicht die Arbeit antun wollen, die Teiler von  $c$  zu suchen. Wir gehen daher alle ganze Zahlen von  $-|c|$  bis  $|c|$  in einer zweiten while-Schleife durch.
- Eine for-Schleife berechnet die Koeffizienten von  $Q(x)$ :  $q[i]=p[i]+a*q[i-1]$ .
- Das letzte Element von  $q$  ist der Rest  $R$ . Wenn dieser von Null verschieden ist, probieren wir es mit der nächsten Hornerzahl  $a$ . Anderenfalls erledigen wir Dreierlei: der Faktor  $(x - a)$  wird ausgegeben,  $P(x)$  wird durch  $Q(x)$  ersetzt (natürlich nur die Koeffizienten) und  $a$  wird zur neuen Konstanten gemacht.
- Sobald die while-Schleife durchgearbeitet ist und kein neuer Teiler  $(x - a)$  gefunden wurde, soll das Programm den verbleibenden Faktor ausgeben und wir beginnen wieder mit einer Abfrage für das nächste Polynom (nach einer Leerzeile).

Das Programm ist möglicherweise ein wenig unbefriedigend, da es nur „brutal“ abgebrochen werden kann. Mit der Zeile `if p[0]==0:break` als dritte Zeile kannst du einen geordneten Abbruch durch die Eingabe von 0 als Koeffizient erzwingen.

Das Horner-Schema wird auch dazu verwendet, um den Funktionswert einer Polynomfunktion  $y = f(x)$  an einer Stelle  $x = a$  zu berechnen. Dazu wird der Algorithmus nur einmal angewendet und der oben als  $R$  angezeigte Wert ist der Funktionswert  $f(a)$ .

Du kannst `horner.py` so abändern, dass eine Ausgabe erfolgt, wie angezeigt:



```
>>>#Running Horner1.py
>>>from Horner1 import *
Koeffizienten: 1,1,-4,-4
Argument: 3.5
37.125
>>>
Koeffizienten: 2,0,0,-5
Argument: -1.2
-0.8528000000000002
>>>
Koeffizienten:
```

Triff auch Vorsorge, dass du Dezimalzahlen eingeben kannst! (siehe Programm 2 in den Programmiergrundlagen)

Hinweis: Auf dem TI-84+ kann man die Koeffizienten als Liste, d.h. zwischen [ ] eingeben. Auf dem PC funktioniert das auch. Auf dem Handheld lassen sich aber die eckigen Klammern über die Ctrl-Taste weder mit `ctrl` `[` noch mit `ctrl` `]` eingeben. Im Editor und in der Shell ist das kein Problem. Nur im Verlauf eines Programms, – wie bei der Antwort auf eine `input`-Abfrage –, geht das leider nicht.

Ich muss mich bei Veit Berger herzlich für den Ausweg (siehe Programmzeile 2) aus dieser misslichen Lage bedanken.