2. Linearfaktordarstellung

2. Linearfaktordarstellung

Beispiel:

$$f(x) = \frac{1}{2}(x-1)(x+2)(x+3) \qquad\qquad g(x) = \frac{1}{2}x^3 + 2x^2 + \frac{x}{2} - 3$$

Aufbau Produkt von Linearfaktoren Polynom

Definition:

Bestehen Faktoren einer Multiplikation aus linearen Elementen der Form $ax + b, \quad a, b \in \mathbb{R}$, so bezeichnet man die Faktoren als Linearfaktoren.

Schauen wir uns die Funktionsgraphen der Fuktion f und g an:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatter

# Defintionsmenge und Funktion
# ----------------------------
a= -5.1 # untere x-Intervallgrenze
b= 5.1 # obere x-Intervallgrenze
c = -5.1# untere y-Intervallgrenze
d = 5.1 # obere y-Intervallgrenze
x = np.linspace(a, b,1000)
y1=0.5*(x-1)*(x+2)*(x+3)
y2=0.5*x**3+2*x**2+0.5*x-3

# ----------------------------
```

```
# Einstellung des Graphen
fig=plt.figure(figsize=(14,14))
ax= fig.add_subplot(2,2,1,aspect=1)
ax1= fig.add_subplot(2,2,2,  aspect=1)


# Definiton der Haupteinheiten, reele Zahlen ohne die 0
def major_tick(x, pos):
    if x==0:
        return ""
    return int(x)

# Achsenskalierung
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.xaxis.set_minor_locator(AutoMinorLocator(2))
ax.yaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_minor_locator(AutoMinorLocator(2))
ax.xaxis.set_major_formatter(FuncFormatter(major_tick))
ax.yaxis.set_major_formatter(FuncFormatter(major_tick))

ax1.xaxis.set_major_locator(MultipleLocator(1))
ax1.xaxis.set_minor_locator(AutoMinorLocator(2))
ax1.yaxis.set_major_locator(MultipleLocator(1))
ax1.yaxis.set_minor_locator(AutoMinorLocator(2))
ax1.xaxis.set_major_formatter(FuncFormatter(major_tick))
ax1.yaxis.set_major_formatter(FuncFormatter(major_tick))


# Position der Achsen im Schaubild
ax.spines[['top','right']].set_visible(False)
ax.spines[['bottom','left']].set_position('zero')

ax1.spines[['top','right']].set_visible(False)
ax1.spines[['bottom','left']].set_position('zero')

# Pfeile für die Achsen
ax.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax.get_yaxis_transform(), cli
ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis_transform(), cli

ax1.plot((1),(0), ls="", marker= ">", ms=7, color="k", transform=ax1.get_yaxis_transform(), c
ax1.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax1.get_xaxis_transform(), c
```

```python
# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
#ax.set_ylabel("f(x)", loc="top", rotation=0)

ax1.set_xlim(a,b)
ax1.set_ylim(c, d)
ax1.set_xlabel("x", loc="right")
#ax1.set_ylabel("f(x)", loc="top", rotation=0)

# Kästchen
ax.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

ax1.grid(linestyle="-", which="major",linewidth=0.7, zorder=-10)
ax1.grid(linestyle="-", which="minor",linewidth=0.5, zorder=-10)

ax.set_title("$f(x)$")
ax1.set_title("$g(x)$")

# Plot der Funktion

ax.plot(x,y1, zorder=10)
ax1.plot(x,y2, zorder=10)

#plt.show()
```

2_Linearfaktordarstellung_files/figure-pdf/cell-2-output-1.png

Satz 1:

1. Jede ganzrationale Funktion in Lienarfaktordarstellung lässt sich durch ausmultiplizieren in eine Polynomfunktion umformen.

2. Nicht jede ganzrationale Funktion besitzt eine Liearfaktordarstellung.

Beweis:

1. klar

2. Gegenbeispiel: $f(x) = x^2 + 1 \quad \square$

Satz 2:

Ist eine ganzrationale Funktion f vom Grad n und der Nullstelle c gegeben, so gibt es eine ganzrationale Funktion g vom Grad n-1, so dass gilt:

$$f(x) = (x - c) \cdot g(x)$$

Beweis:

Gegeben: - ganzrationale Funktion f - grad$(f) = n$ - c Nullstelle von f

Verschiebung der Nullstelle c in den Ursprung durch Verschieben des Graphen von f entlang der x-Achse um $-c$ erzeugt eine neue Funktion h:

$$h(x) = f(x - (-c)) = f(x + c)$$

Eigenschaften von h: - h ist auch eine ganzrationale Funktion mit $h(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_0$ - $x_0 = 0$ ist eine Nullstelle von h, d.h. $h(0) = f(0 + c) = f(c) = 0$

Es folgt damit: $a_0 = 0$

und es gilt:

$$\begin{aligned} h(x) &= a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x \\ &= x \cdot \left(a_n x^{n-1} + a_{n-1}x^{n-2} + ... + a_1\right) \\ &= x \cdot k(x) \end{aligned}$$

Eigneschaften von k(x): - ganzrationale Funktion - Gleichung $k(x) = a_n x^{n-1} + a_{n-1}x^{n-2} + ... + a_1$ - \$grad(k)=n-1 \$

Zurückverschiebung des Graphen von h um c entlang der x-Achse ergibt den Graphen von f und es gilt:

$$\begin{aligned} f(x) &= h(x - c) \\ &= (x - c) \cdot k(x - c) \\ &= (x - c) \cdot g(x), \quad \text{mit } g(x) = k(x - c) \end{aligned}$$

mit grad$(g) =$ grad$(k) = n - 1\square$

Satz 3:

Eine ganzrationale Funktion f vom Grad n hat höchstens n Nullstellen.

Beweis:

Gegeben:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

mit $\text{grad}(f) = n$

$\Rightarrow$ Es gibt ein c_1, so das gilt:

$$\begin{aligned} f(x) &= (x - c_1) \cdot \left(a_{n-1} x^{n-1} + ... + a_1 x + a_0\right) \\ &= (x - c_1) \cdot g(x) \end{aligned}$$

$\Rightarrow \quad \text{grad}(g) = \text{grad}(f) - 1 = n - 1$

Sei c_2 weiter Nullstelle von f(x).
$\Rightarrow$

$$\begin{aligned} f(x) &= (x - c_1)(x - c_2) \left(a_{n-2} x^{n-2} + ... + a_1 x + a_0\right) \\ &= (x - c_1)(x - c_2) \cdot h(x) \end{aligned}$$

$\Rightarrow \quad \text{grad}(h) = \text{grad}(f) - 2 = n - 2$

Die Durchführung dieser Schritte ist insgesamt maximal n-mal möglich. $\square$