

3 Graphen der Exponentialfunktion

Frage: Welches Verhalten zeigen Funktionen, welche die e-Funktion enthalten für $x \rightarrow \infty$?

Beispiele:

1. $f(x) = e^x$
2. $g(x) = e^{-x}$
3. $h(x) = x \cdot e^{-x}$

```
In [5]: import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatter

# Definitionsmenge 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 = np.exp(x)
y2 = x * np.exp(-x)
y3 = np.exp(-x)
# -----

# Einstellung des Graphen
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(1, 1, 1, aspect = 1)

# Definition der Haupteinheiten, reelle 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))

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

# Pfeile für die Achsen
ax.plot((1), (0), ls=":", marker=">", ms=7, color="k", transform=ax.get_yaxis
```

```

ax.plot((0),(1), ls="", marker= "^", ms=7, color="k", transform=ax.get_xaxis

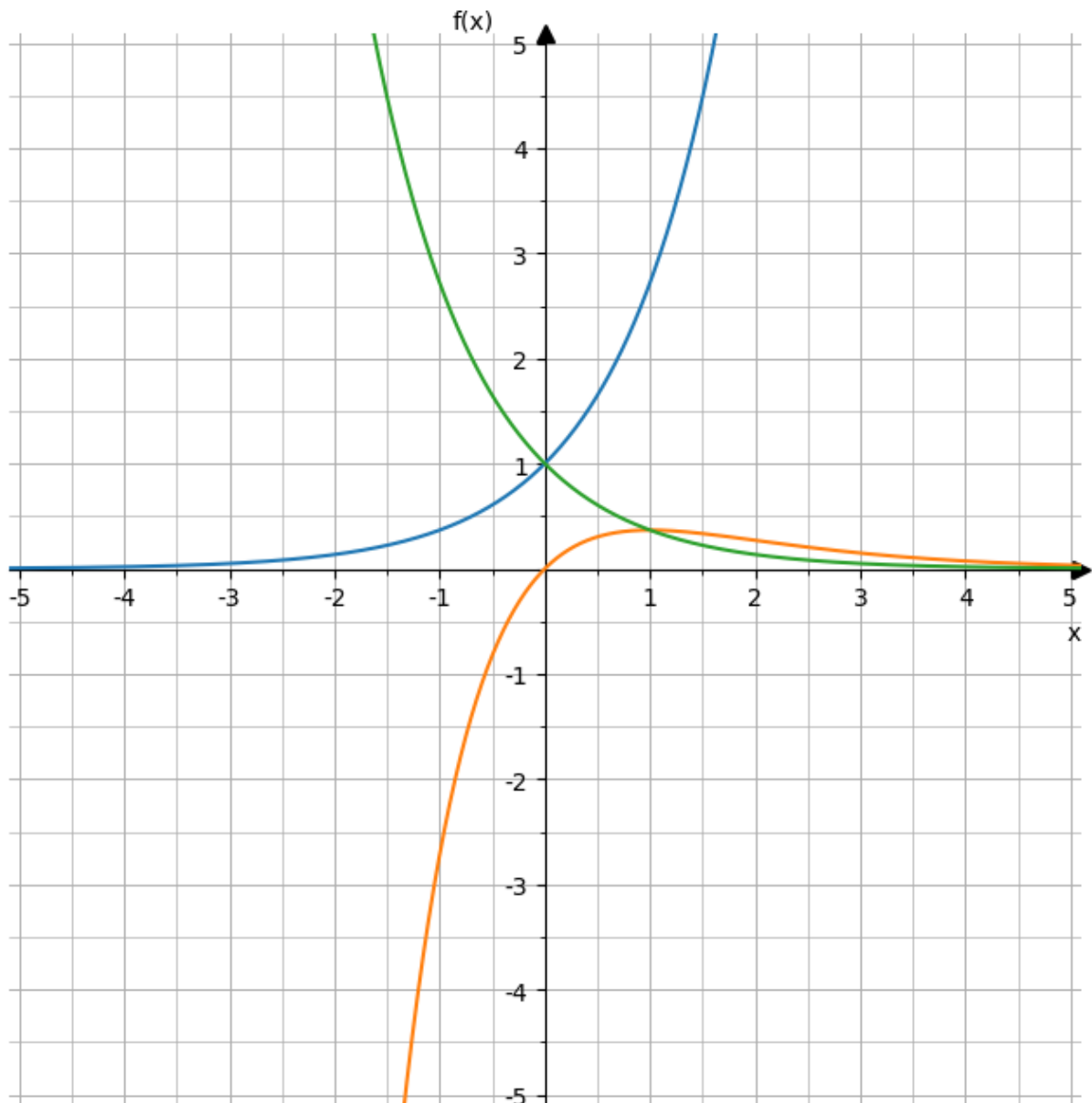
# Achsenlänge und Beschriftung
ax.set_xlim(a,b)
ax.set_ylim(c, d)
ax.set_xlabel("x", loc="right")
ax.set_ylabel("f(x)", loc="top", rotation=0)

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

# Plot der Funktion
ax.plot(x,y1, zorder=10)
ax.plot(x,y2, zorder=10)
ax.plot(x,y3, zorder=10)
plt.show()

```

Out[5]: [<matplotlib.lines.Line2D at 0x1165ba450>]



Beobachtung: Bei der Funktion h bestimmt e^x das Verhalten für $x \rightarrow \pm\infty$

Satz:

Bei Funktionen der Form

- $f(x) = x^n \cdot e^{a \cdot x}$
- $f(x) = x^n \cdot e^{-a \cdot x}$

mit $a > 0$

bestimmt der Faktor, welcher die e-Funktion enthält, das Verhalten von $f(x)$ für $x \rightarrow \infty$.

Der Faktor x^n bestimmt das Vorzeichen.

Beweis:

...

Beispiel:

Extrempunkt bestimmen von $f'(x) = (x^2 - 3) \cdot e^x$.

1. Ableitungen:

$$f'(x) = 2x \cdot e^x + (x^2 - 3) \cdot e^x = (x^2 + 2x - 3) \cdot e^x$$

$$f''(x) = (2x + 2) \cdot e^x + (x^2 - 2x - 3)e^x = (x^2 - 3) \cdot e^x$$

2. Extremstellen

A. Notwendige Bedingung:

$$\begin{aligned} f(x) &= 0 \\ x^2 + 2x - 3 &= 0 \\ x_{1,2} &= -1 \pm \sqrt{1 - (-3)} \\ x_1 &= 1 \quad x_2 = -3 \end{aligned}$$

B. Hinreichende Bedingung:

$$f''(1) < 0 \Rightarrow \text{Hochpunkt}$$

$$f''(1) > 0 \Rightarrow \text{Tiefpunkt}$$

3. Koordinaten des Punktes berechnen:

$$P(1|f(1)) = P(1|-2e)$$

4. Zeichnung:

```
In [4]: import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatter

# Definitionsmenge und Funktion
# -----
a = -5.1 # untere x-Intervallgrenze
b = 5.1 # obere x-Intervallgrenze
c = -6.1 # untere y-Intervallgrenze
d = 5.1 # obere y-Intervallgrenze
x = np.linspace(a, b, 1000)
y1 = (x**2 - 3) * np.exp(x)
# -----

# Einstellung des Graphen
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(1, 1, 1, aspect = 1)

# Definition der Haupteinheiten, reelle 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))

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

# Pfeile für die Achsen
ax.plot((1), (0), ls="none", marker=">", ms=7, color="k", transform=ax.get_yaxis)
ax.plot((0), (1), ls="none", marker="^", ms=7, color="k", transform=ax.get_xaxis)

# 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)

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

# Plot der Funktion
ax.plot(x, y1, zorder=10)
plt.show()
```

Out[4]: [

