5. Stammfunktionen und ihre Graphen

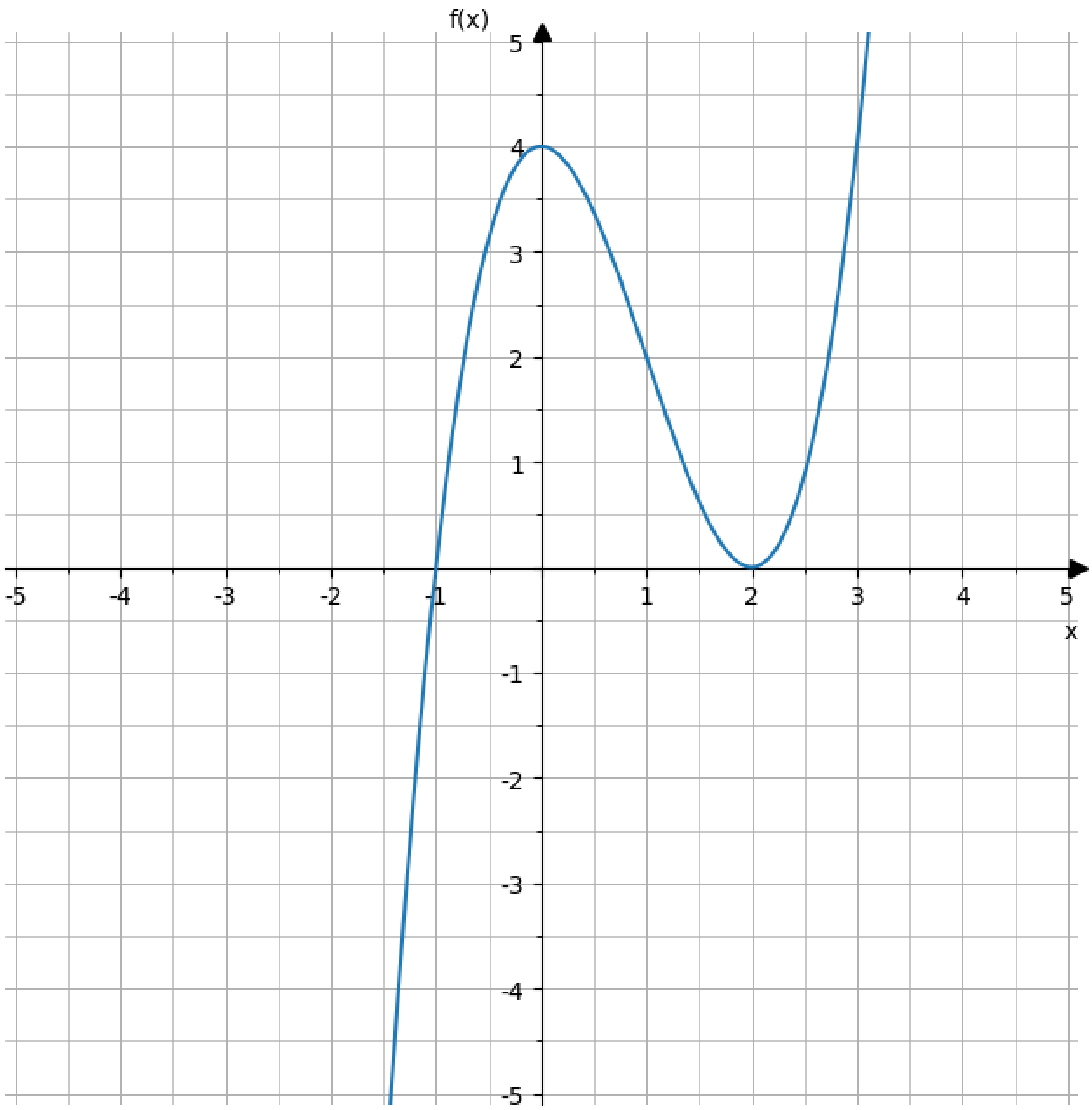
Analog dem graphischen Ableiten kann man bei bekanntem Graphen einer Funktion f eine Stammfunktion bestimmen. Dazu benutzt man charakteristische Punkte

In [32]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# 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= (x+1)*(x-2)**2
y2=x**3-3*x**2+4
# ----------------------------

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

# 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)
#plt.show()
```

Out[32]:

```
[<matplotlib.lines.Line2D at 0x11f952ed0>]
```



Nullstellen von f bedeuten für F

- $N_1(-1|0)$ und $N_2(2|0)$ sind die Nullpunkte
- n_1 ungerade Nullstelle, mindestens also eine einfache Nullstelle
- n_2 gerade Nullstelle, mindestens also eine doppelte Nullstelle
- bei N_1 macht der Graph K von f einen Vorzeichenwechsel
- bei N_2 macht der Graph K von f keinen Vorzeichenwechsel

$\Rightarrow$ F hat an der Stelle n_1 ein Minimum
$\Rightarrow$ F hat an der Stelle n_2 eine Sattelstelle

Extremstellen von f bedeuten für F

- $E_1(0|4)$ und $E_2(2|0)$ sind die Extrempunkte
- Es gilt an den Extremstellen (notwendige Bedingung) dass $f'(e_1) = f'(e_2) = 0$
- Es gilt damit auch, $f''(e_1) = f''(e_2) \neq 0$ (hinreichende Bedingung)
- f' entdpricht F'' und damit gilt: $F''(e_1) = F''(e_2) = 0$

$\Rightarrow$ F hat an den Extremstellen von f jeweils eine Wendestelle

Vorzeichen von f

- Für $(-\infty; -1)$ gilt: $f(x) < 0$
  $\Rightarrow$ F ist auf $(-\infty; -1)$ **streng monoton fallend**
- Für $(-1; \infty)$ gilt: $f(x) \geq 0$
- Für $(-1; \infty)$ bis auf $x = 2$ gilt: $f(x) > 0$
  $\Rightarrow$ F ist auf $(-1; \infty)$ **streng monoton steigend**. Die eine Stelle $x = 2$, an der gilt: $f(2) = 0$, ändert an der Monotonie nichts. (vgl. $f(x) = x^3$)

Ergebnisse für eine Skizze zusammentragen

1. Trage den Tiefpunkt von Fin ein Kooridnatensystem ein. Der y-Wert des Graphen ist nicht bekannt, kann also beliebig gewählt werden.
2. Ergänze darauf die weiteren Eingenschaften.
3. Konkretisierung: Bestimme ausgehdned von einem Punkte die Differenz der beiden y-Werte zu einem anderen Punkt.
   Beispiel:

$$\int\limits_{-1}^{0} f(x)dx = F(0) - F(-1)$$

   Kästchen abzählen ergibt:

$$F(0) - F(-1) \approx \frac{10}{4} = 2,5$$

   d.h.

$$F(0) \approx F(-1) + 2,5$$

   damit ist der y-Wert an der Stelle $x = 0$ um 2,5 Einheiten größer als der y-Wert an der Stelle $x = -1$.

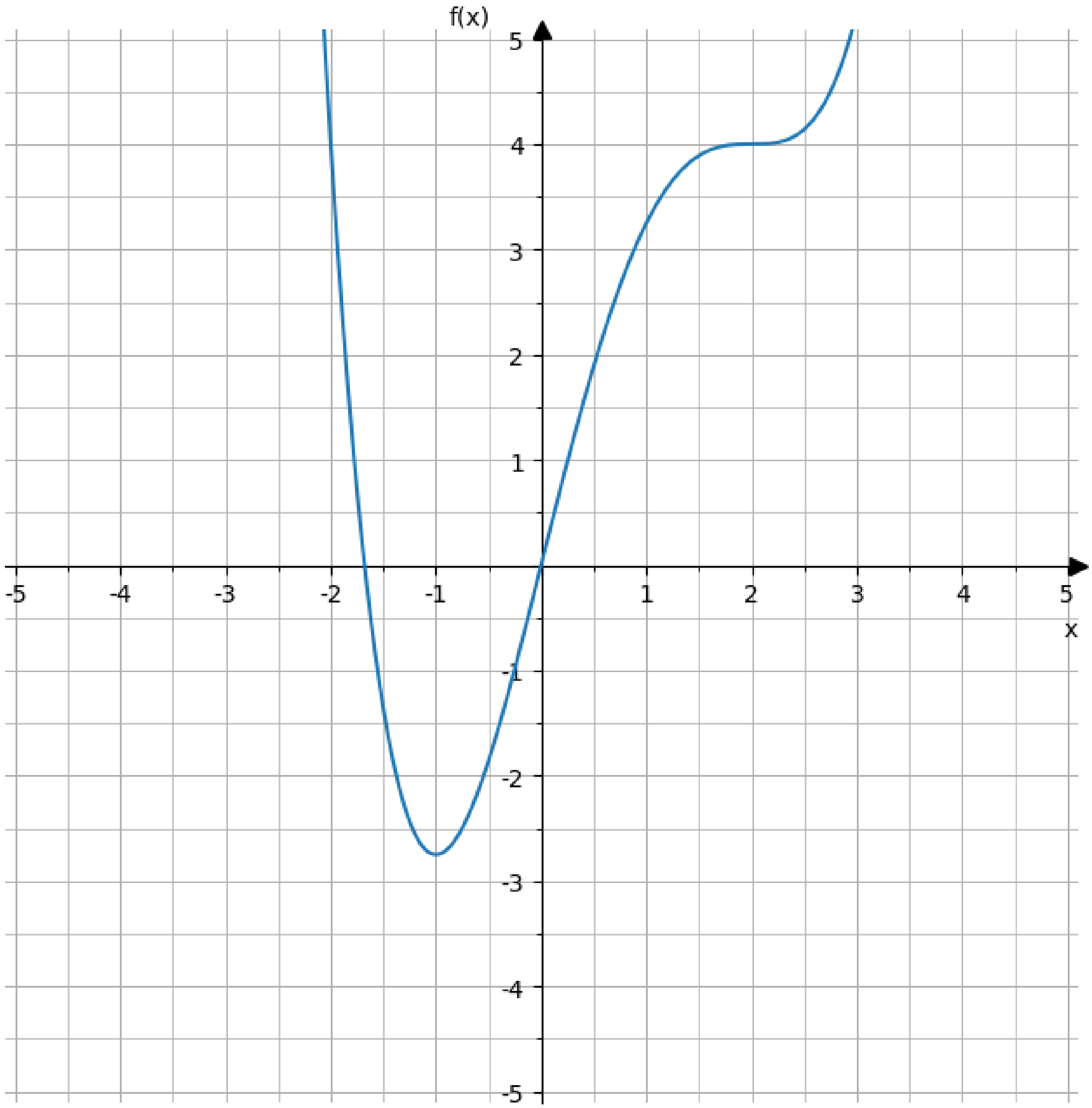
Bemerkung: Jede Funktion $G(x) = F(x) + c$ mit $c \in \mathbb{R}$ ist eine weitere Stammfunktion von f. Daher ist jede Verschiebung des Graphen auch ein Graph einer Stammfunktion von f.

In [34]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import AutoMinorLocator, MultipleLocator, FuncFormatt

# 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= 0.25*(x**4)-x**3+4*x
# -----------------------------

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

# 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)
#plt.show()
```

Out[34]:

```
[<matplotlib.lines.Line2D at 0x11fa9a910>]
```



In [ ]: