

Plot W-Funktion mit SGD-Verfahren

```
In [6]: # Libraries und Initialisierung
import time
import numpy as np
import matplotlib.pyplot as plt

# Definition der Funktion und ihrer Ableitung
def f(x):
    return x**4 - 4*x**2 + x + 3

def f_prime(x):
    return 4*x**3 - 8*x + 1

# SGD-Parameter
learning_rate = 0.01
num_iterations = 1000

# Initialisierung
x = 0.0
x_history = [x]
f_history = [f(x)]
```

```

In [7]: # SGD-Algorithmus
for _ in range(num_iterations):
    gradient = f_prime(x)
    x = x - learning_rate * gradient
    x_history.append(x)
    f_history.append(f(x))

# Plotten der Funktion
x_vals = np.linspace(-2, 2, 400)
y_vals = f(x_vals)

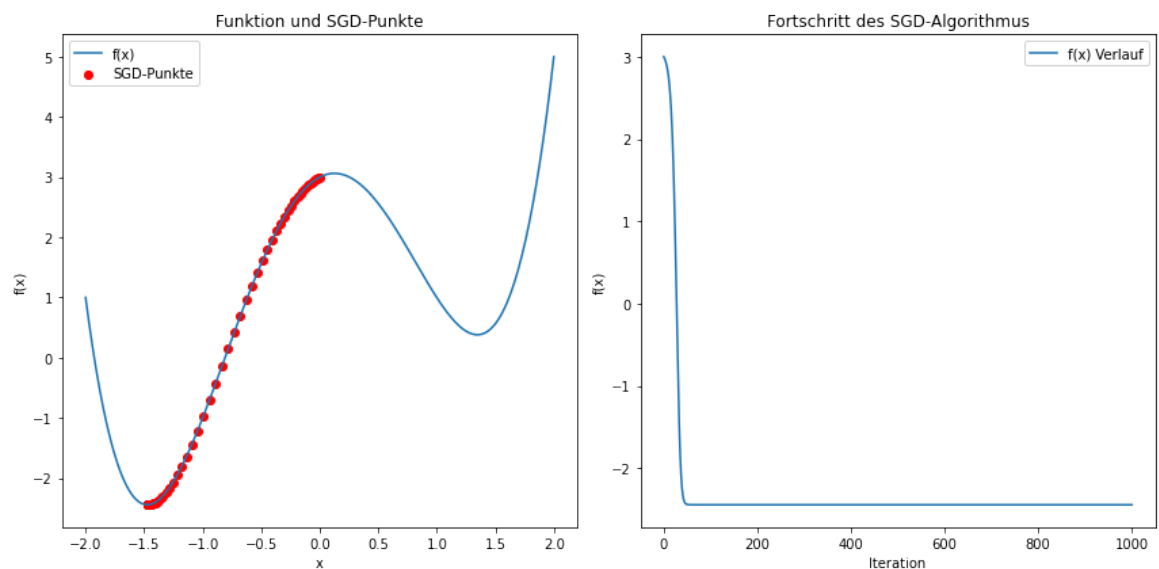
plt.figure(figsize=(12, 6))

# Plot der Funktion
plt.subplot(1, 2, 1)
plt.plot(x_vals, y_vals, label='f(x)')
plt.scatter(x_history, [f(x) for x in x_history], color='red', label='SGD-Punkte')
plt.title('Funktion und SGD-Punkte')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()

# Plot des Fortschritts
plt.subplot(1, 2, 2)
plt.plot(f_history, label='f(x) Verlauf')
plt.title('Fortschritt des SGD-Algorithmus')
plt.xlabel('Iteration')
plt.ylabel('f(x)')
plt.legend()

plt.tight_layout()
plt.show()

```



Ergebniss

Nach ungefähr 150 Iterationen ergibt sich das folgende Minimum: Das minimale x : $x_{\min} \approx -1.473$ Der minimale Funktionswert: $f(x_{\min}) \approx -2.444$ Die Iterationen enden, wenn der Betrag der Änderung in x kleiner als die Toleranzschwelle wird. Dies hängt in diesem Fall aber auch stark am Startpkt ab. Hätte man eine Startpunkt etwa bei $x=1.5$ gewählt wäre man beim lokalen Minimum bei $x = 1.36$ gelandet....\

```
In [8]: # print current date and time

print("Date & Time:",time.strftime("%d.%m.%Y %H:%M:%S"))
# end of import test
print ("*** End of SGD Beispiel ***")
```

```
Date & Time: 18.09.2024 14:18:28
*** End of SGD Beispiel ***
```