1.2

In [355…
```python
# import required modules

from typing import Callable

import matplotlib.pyplot as plt
import numpy as np
import numpy.typing as npt
```

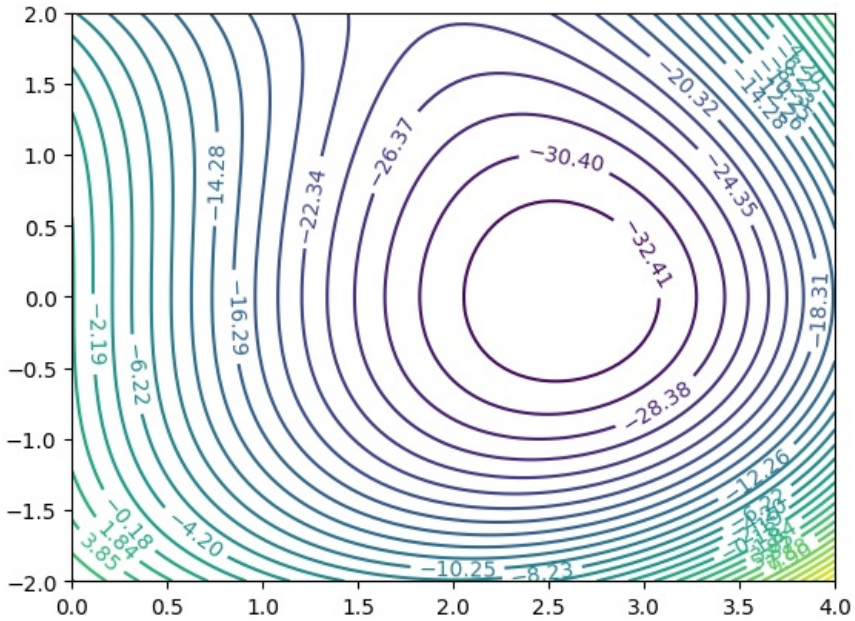
In [356…
```python
# define ranges (min, max, steps)
RANGE_X1 = (0, 4, 100)
RANGE_X2 = (-2, 2, 100)
# RANGE_X1 = (-100000, 100000, 1000)
# RANGE_X2 = (-100000, 100000, 1000)
```

In [357…
```python
# function from the problem
def function2(x: npt.ArrayLike) -> float:
    return pow(x[0], 3) + 2*x[0]*pow(x[1], 2) - pow(x[1], 3) - 20*x[0]
```

In [358…
```python
# another function to prepare the data that is going to be plotted
def prep_data(function: Callable[[float, float], float], range_x1: tuple[float, float, float], range_x2: tuple[
    x1 = np.linspace(range_x1[0], range_x1[1], range_x1[2])
    x2 = np.linspace(range_x2[0], range_x2[1], range_x2[2])
    x1, x2 = np.meshgrid(x1, x2)
    fx = function([x1, x2])
    return x1, x2, fx
```

In [359…
```python
# simple function to plot data
def plot_data(x1: npt.ArrayLike, x2: npt.ArrayLike, fx: npt.ArrayLike) -> None:
    _, ax = plt.subplots()
    levels = np.linspace(np.min(fx), np.max(fx), 30)
    CS = ax.contour(x1, x2, fx, levels=levels)
    ax.clabel(CS, inline=True, fontsize=10)
    plt.show()
```

In [360…
```python
# main function so this can run outside of a jupyter notebook
if __name__ == "__main__":
    x1, x2, fx = prep_data(function2, RANGE_X1, RANGE_X2)
    plot_data(x1, x2, fx)
    # print(f"min from sampling {np.min(fx)}")
    # print(f"x1: {x1[np.where(fx == np.min(fx))]}")
    # print(f"x2: {x2[np.where(fx == np.min(fx))]}")
```



local minimum = -34 at x1 = 2.5 and x2 = 0

global minimum does not seem to exist as the function keeps reducing as x1 reduces and x2 increases