In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

inital

In [ ]:
```
a = 2
b = 3
x0 = (a, 1)
epsilon = 1e-15
```

In [ ]:
```
def f(x):
    x1 = x[0]
    x2 = x[1]
    return 0.5*(x1**2 + (a * x2**2)) - np.cos(b * (x1 + x2))
```

In [ ]:
```
def f_gradient(x):
    x1 = x[0]
    x2 = x[1]
    return np.array([(x1 + np.sin(b*(x1+x2))*b),
                     (a*x2 + np.sin(b*(x1+x2))*b)])
```

steepest-descent method

In [ ]:
```
def steepest_descent(x, max_iterations, t):
    x_hist = []
    x_hist.append(x)
    results = []
    results.append(f(x))
    k = 1
    while True:
        x = x + (t * -f_gradient(x))
        x_hist.append(x)
        results.append(f(x))
        if k == max_iterations or abs(results[-1] - results[-2]) <= epsilon:
            break
        else:
            k += 1
    return x_hist, results
```

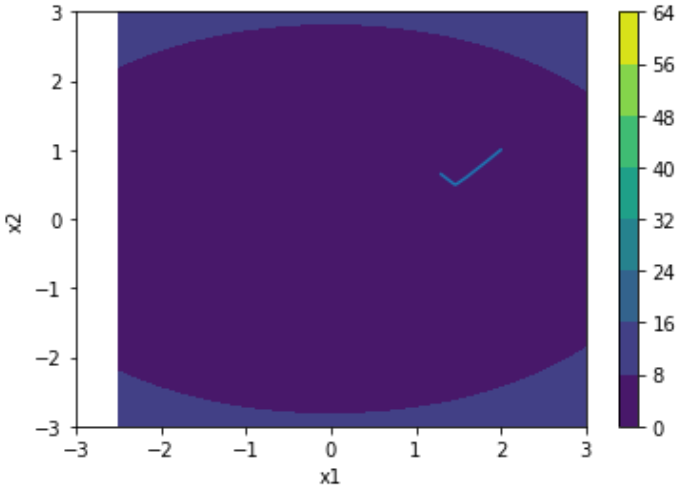
In [ ]:
```
def plot_result(x):
    x1_plot = [float(x[0]) for x in x]
    x2_plot = [float(x[1]) for x in x]

    xlist = np.linspace(-2.5, 10.0, 100)
    ylist = np.linspace(-3.5, 3.5, 100)
    X, Y = np.meshgrid(xlist, ylist)
    Z = 1/2 * (X**2 + (a * Y**2))
    fig, ax = plt.subplots(1,1)
    cp = ax.contourf(X, Y, Z)
    fig.colorbar(cp)
    ax.set_xlabel('x1')
    ax.set_ylabel('x2')
    ax.set_xlim(-3,3)
    ax.set_ylim(-3,3)
    ax.plot(x1_plot, x2_plot)
    plt.show()
```

find x, *p* by steepest-descent method

In [ ]:

```
t = 0.05
max_iterations = -1
x_hist,f_SD_t1 = steepest_descent(x0,max_iterations,t)
p_opt = min(f_SD_t1)
print('p* =',p_opt)
print('x_k',x_hist[:5])
print('k iterations:', len(x_hist))
# create csv file
df = pd.DataFrame(x_hist)
df.to_csv('problem3_t1.csv', index=False, header=False)
plot_result(x_hist)
```

```
p* = 0.3597727553131582
x_k [(2, 1), array([1.83818223, 0.83818223]), array([1.59856705, 0.60665794]), array([1.4696
791 , 0.49703254]), array([1.45225787, 0.50339202])]
k iterations: 195
```



In [ ]:

```
t = 0.15
max_iterations = 4000
x_hist,f_SD_t1 = steepest_descent(x0,max_iterations,t)
p_opt = min(f_SD_t1)
print('p* =',p_opt)
print('x_k',x_hist[:5])
print('k iterations:', len(x_hist))

# create csv file
df = pd.DataFrame(x_hist)
df.to_csv('problem3_t2.csv', index=False, header=False)
plot_result(x_hist)
```

```
p* = 0.404500756569134
x_k [(2, 1), array([1.51454668, 0.51454668]), array([1.37495921, 0.44777721]), array([1.4961
8446, 0.64091317]), array([1.21426593, 0.39114837])]
k iterations: 4001
```