ld-opgaver

Magnus Chr. Hvidtfeldt

Technical University of Denmark, Lyngby, DK,

s255792@dtu.dk

```
from sympy import *
init_printing()
from dtumathtools import *
x1, x2 = symbols('x1 x2')
```

1 Kvadratiske former

1.1 Spg a

For $q_1(x_1, x_2)$ we have

```
A = Matrix([[11, -12],
            [-12, 4]])
b = Matrix([-20, 40])
x = Matrix([x1, x2])
c = -60

P1 = x.T * A * x
P2 = b.T * x
q1 = simplify(P1[0] + P2[0] + c)
simplify(P1[0] + P2[0] + c)
```

$11x_1^2 - 24x_1x_2 - 20x_1 + 4x_2^2 + 40x_2 - 60$

For $q_2(x_1, x_2)$ we have

```
A = Matrix([[11, 0],
            [-24, 4]])
b = Matrix([-20, 40])
x = Matrix([x1, x2])
c = -60

P1 = x.T * A * x
P2 = b.T * x
```

```
simplify(P1[0] + P2[0] + c)
```

$x_1\left(11x_1 - 24x_2\right) - 20x_1 + 4x_2^2 + 40x_2 - 60$

For $q_3(x_1, x_2)$ we have

```
A = Matrix([[73/5, -36/5],
            [-36/5, 52/5]])
b = Matrix([-44, 8])
x = Matrix([x1, x2])
c = -60

P1 = x.T * A * x
P2 = b.T * x
q3 = simplify(P1[0] + P2[0] + c)
simplify(P1[0] + P2[0] + c)
```

$14.6x_1^2 - 14.4x_1x_2 - 44.0x_1 + 10.4x_2^2 + 8.0x_2 - 60.0$

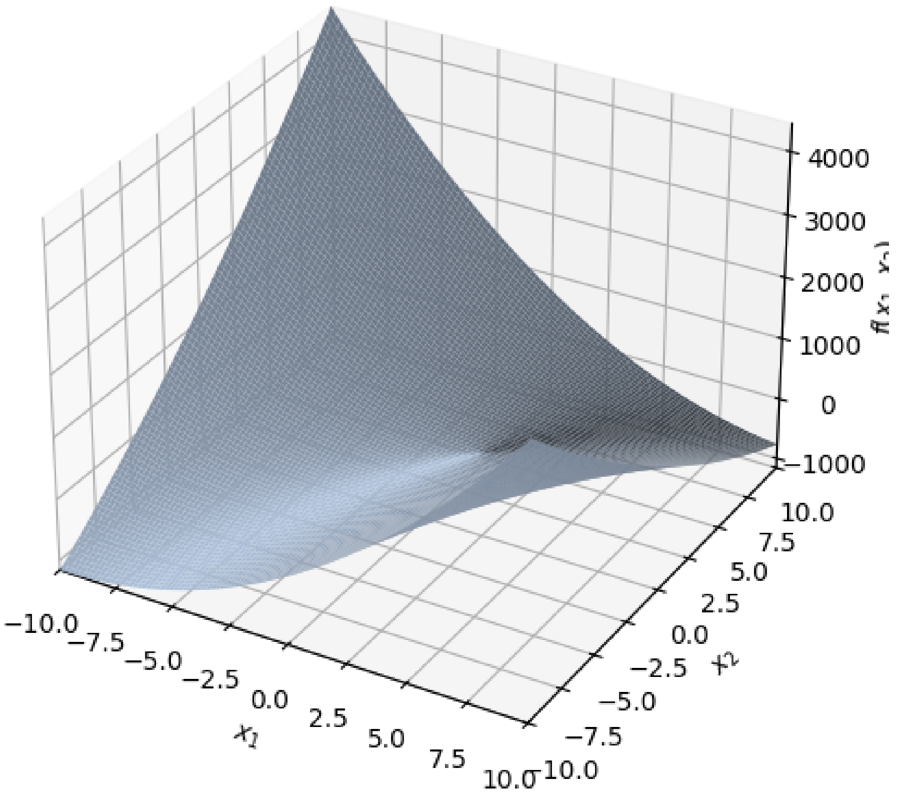
1.2 Spg b

Nej, da $q_1 = q_2$

1.3 Spg c

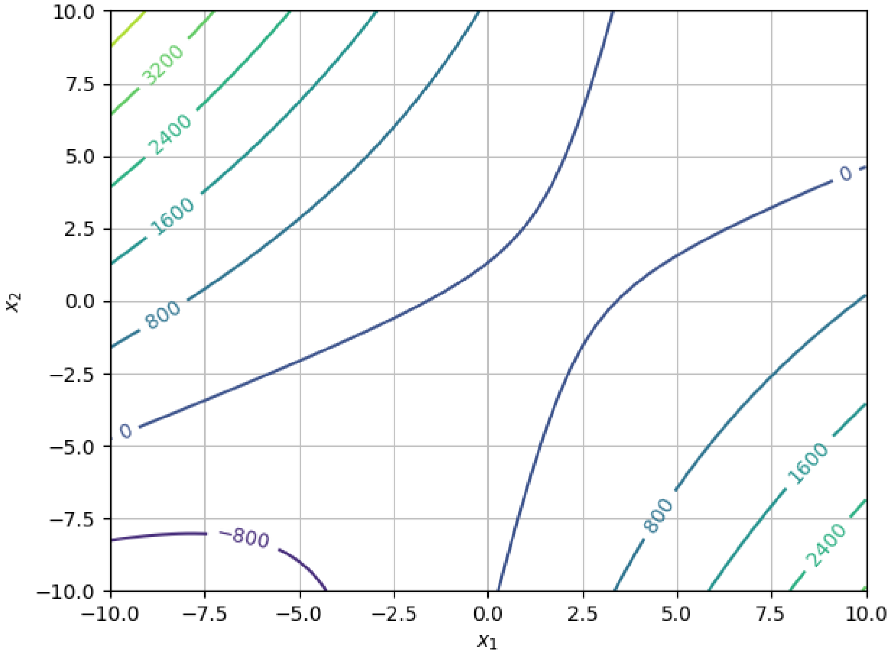
Plot af q_1

```
dtuplot.plot3d(q1, rendering_kw = {"color": "lightsteelblue"})
```



```
<spb.backends.matplotlib.matplotlib.MatplotlibBackend at 0x129a0d580>
```

```
dtuplot.plot_contour(q1, levels=[0], fill=False)
```

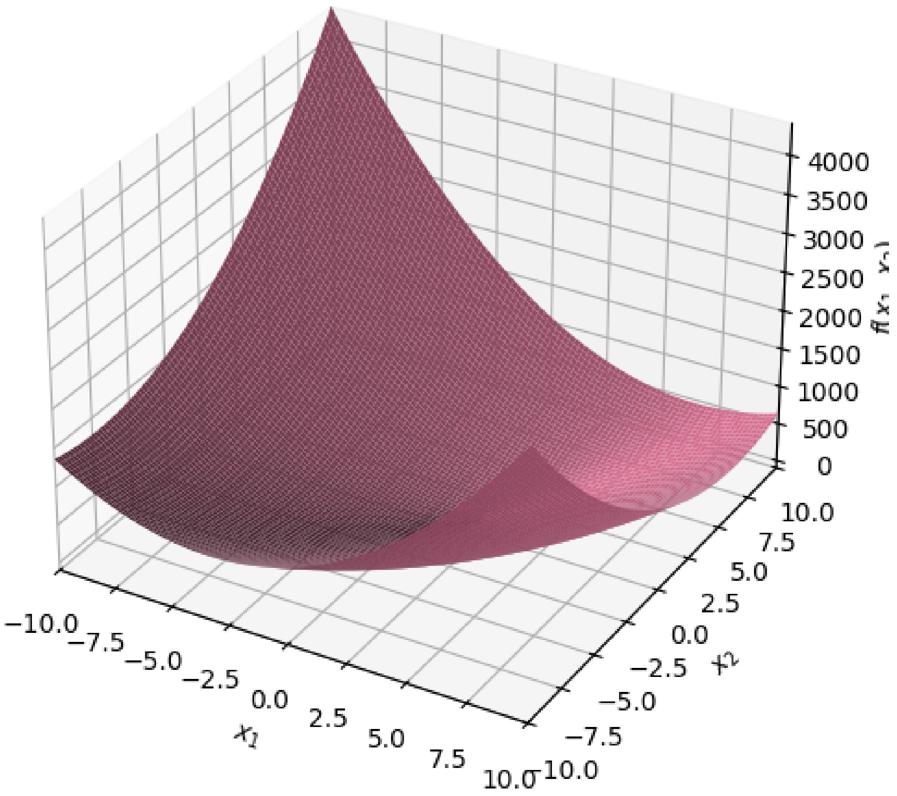


```
<spb.backends.matplotlib.matplotlib.MatplotlibBackend at 0x12baa7b60>
```

The graph has a saddle sort of shape, because of the negative at the ends. Although it only occurs in the one end, so maybe its not called saddle.

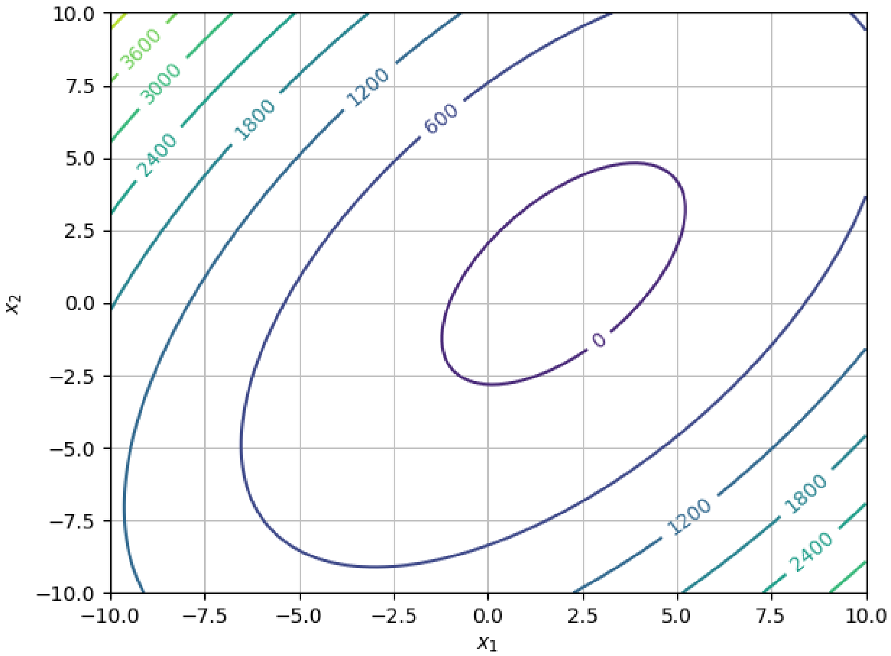
Plot of q_3:

```
dtuplot.plot3d(q3, rendering_kw = {"color": "palevioletred"})
```



```
<spb.backends.matplotlib.matplotlib.MatplotlibBackend at 0x12dfdf650>
```

```
dtuplot.plot_contour(q3, levels=[0], fill=False)
```



```
<spb.backends.matplotlib.matplotlib.MatplotlibBackend at 0x12baa7500>
```

For q_3, we can see that its not a saddle, but a minimal point.

1.4 Spg d

Meaning q_3 is the function with a minimal point, and its laying (2,1). We call it a saddle point.

2 Softmax-funktionen

2.1 Spg a

Calulcate softmax of

```
def softmax(x):
    arr = []
    for i in x:
        nvn = exp(i)
        btm = 0
        for j in x:
            btm += exp(j)

        arr.append((nvn/btm).evalf().round(5))
    display(arr)
```

```
x = [1, 2, -5]
softmax(x)
```

```
[0.26876, 0.73057, 0.00067]
```

```
x = [10, 2, -5]
softmax(x)
```

```
[0.99966, 0.00034, 0.0]
```

```
x = [100, 2, -5]
softmax(x)
```

```
[1.0, 0.0, 0.0]
```

2.2 Spg b

We can see that we greatest value in the input is, the one with the greatest probability at the output. When the difference in the greatest value and the others increase, the probability of that input value increases.

2.3 Spg c

Yes, softmax is continuous since e^x is continuous and so is the operations.

2.4 Spg d

Softmax is not injective. It is not surjective, if we consider the codomain $\mathbb{R}^n$, because it cannot take negative probabilities.