问题3

In [1]:

```
import math
import plotly
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from mitosheet import sheet

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
```

In [ ]:

```

```

In [2]:

```
# 附件参数
copper_ball_material = 'copper'  # 小球材质
copper_ball_R = 10e-3     # m 小球半径
copper_ball_M = 35e-3     # kg 小球质量
mag_sen_rubber_L = 10e-2  # m 磁敏橡胶长
```

In [ ]:

```

```

In [3]:

```
import math
import latexify

@latexify.with_latex
def K(v_0, v_1, E=6, R=copper_ball_R, L=mag_sen_rubber_L):  # latex
    return 202.13 * R / (2.44 * B**2 + 0.14 + E) - 219.6 * R**2 * math.sqrt((v_0**2 - v_1**2) / (L * (2.44 * B**2 + 0.14 + E)**(4/3)))
K
```

Out[3]:

$$\mathrm{K}(v_0, v_1, E, R, L) \triangleq \frac{202.13R}{2.44B^2 + 0.14 + E} - 219.6R^2\sqrt{\frac{v_0^2 - v_1^2}{L(2.44B^2 + 0.14 + E)^{\frac{4}{3}}}}$$

In [ ]:

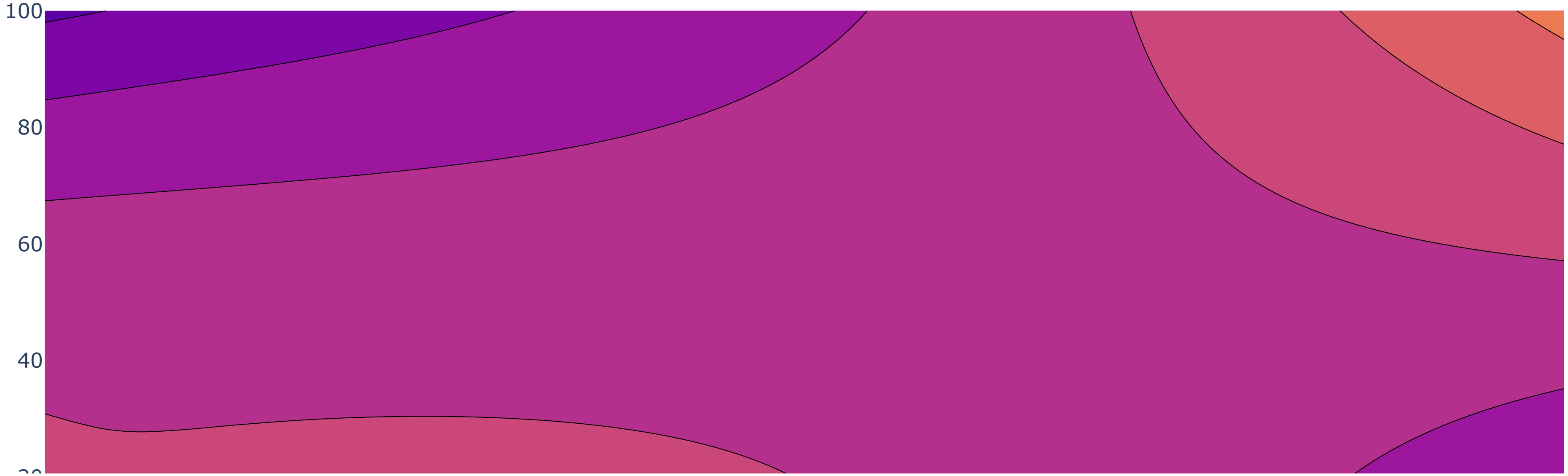
```

```

In [4]:

```
M = 20
```

```
X = np.linspace(0, 200, 201)
Y = np.linspace(0, 100, 101)
xgrid, ygrid = np.meshgrid(X, Y)
addgrid = xgrid**2 + ygrid**2
Z = (xgrid - 100) * (ygrid - 50) * addgrid**0.5
Z = (0.7 - 0.1) * ((Z - Z.min()) / (Z.max() - Z.min())) + 0.1
print(X.shape, Y.shape, Z.shape)
print("max(Z):", Z.max(), ", min(Z):", Z.min())

trace = go.Contour(x=X, y=Y, z=Z)
fig = go.Figure(data=[trace])
fig.show()
fig.write_image("K_distribution.png")
```

```
(201,) (101,) (101, 201)
max(Z): 0.7 , min(Z): 0.1
```



In [ ]:

```

```

In [5]:

```
def get_K(B, E=5, R=copper_ball_R, L=mag_sen_rubber_L, v0_2_sub_v1_2=0.01):  # latex
    return 202.13 * R / (2.44 * B**2 + 0.14 + E) - 219.6 * R**2 * (v0_2_sub_v1_2 / (L * (2.44 * B**2 + 0.14 + E)**(4/3)))**0.5
```

In [ ]:

```

```

In [6]:

```
import math
import latexify

@latexify.with_latex
def y(B):  # latex
```

```
    return 202.13 * R / (2.44 * B**2 + 0.14 + E) - 219.6 * R**2 * math.sqrt((v_0**2 - v_1**2) / (L * (2.44 * B**2 + 0.14 + E)**(4/3))) - K
y
```

Out[6]:

$$y(B) \triangleq \frac{202.13R}{2.44B^2 + 0.14 + E} - 219.6R^2 \sqrt{\frac{v_0^2 - v_1^2}{L(2.44B^2 + 0.14 + E)^{\frac{4}{3}}}} - K$$

In [ ]:

```
 
```

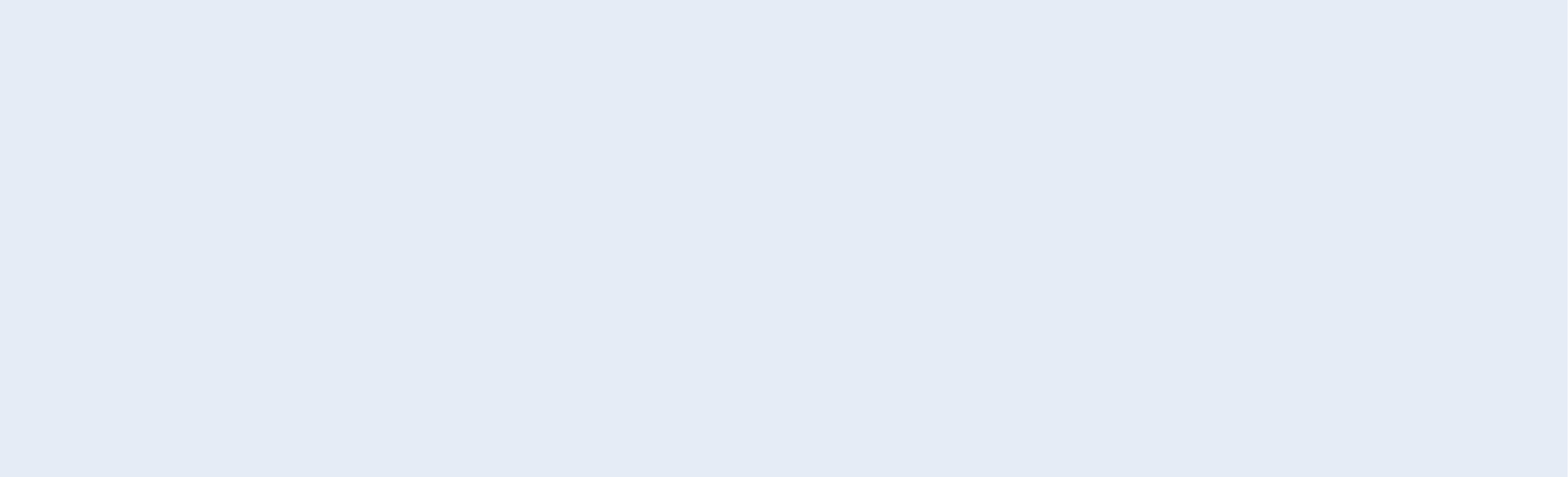
In [7]:

```
def get_B(K):
    from scipy.optimize import fsolve

    def object_func(B, K, E=2, delta_v2=16, R=10e-3, L=10e-2):
        return 202.13 * R / (2.44 * B**2 + 0.14 + E) - 219.6 * R**2 * (delta_v2 / (L * (2.44 * B**2 + 0.14 + E)**(4/3)))**0.5 - K

    B = np.zeros((101, 201))
    for i in range(101):
        for j in range(201):
            B[i, j] = fsolve(object_func, x0=50e-3, args=(K[i, j], ))
    return B
B = get_B(Z)
print(B.shape)
B = np.abs(B)
trace = go.Contour(x=X, y=Y, z=B)
fig = go.Figure(data=[trace])
fig.show()
fig.write_image("B_distribution.png")
```

```
(101, 201)
```

In [8]:

```
E = 2
delta_v2 = 16
B = np.linspace(0, 1000e-3, 1000)
K = get_K(B, E=E, v0_2_sub_v1_2=delta_v2)
fig = go.Figure(data=[go.Scatter(x=B, y=K)])
fig.update_layout(
    title=fr'$弹性模量~E=2~MPa\\初、末速度平方之差 ~ V_0^2 - V_1^2 = {delta_v2}~(m^2 / s^2)$',
    xaxis_title=r"$磁感应强度~B$",
    yaxis_title=r"$滚动摩擦系数 ~ K$",
)
# fig.show()
fig.write_image("B-K.png")
```



In [ ]: