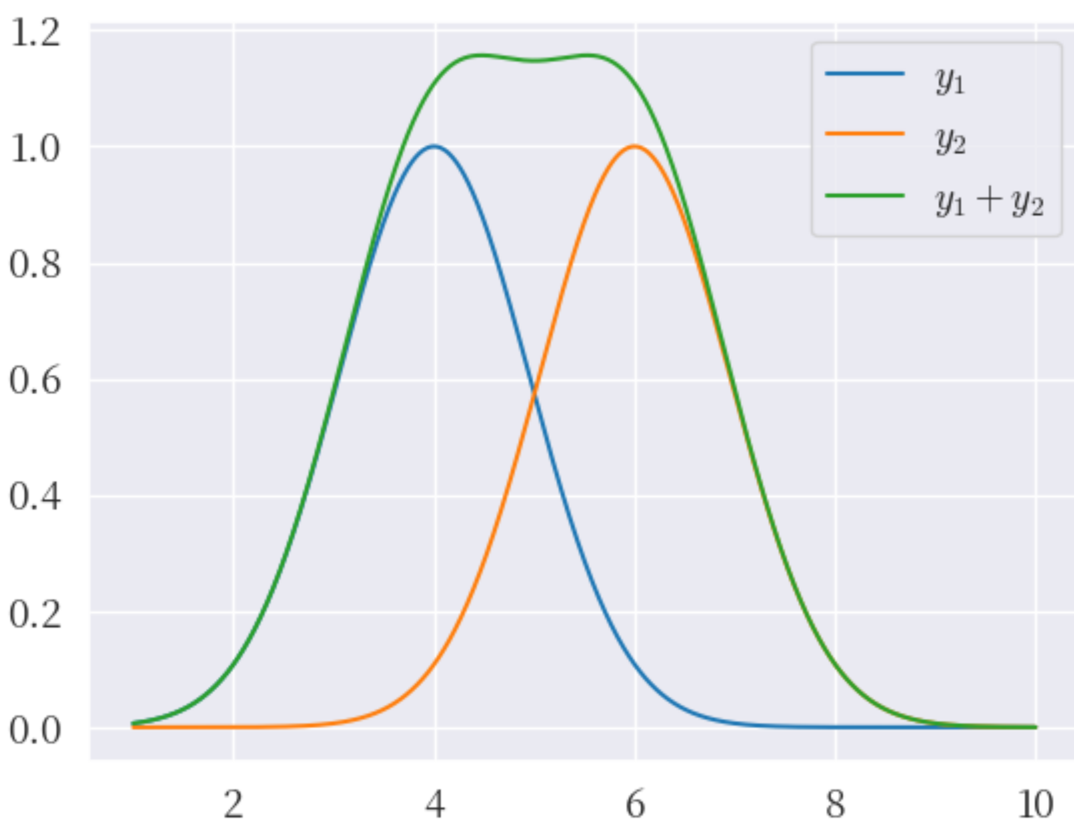
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

目的

求证预乘谱中的双峰形式本质。

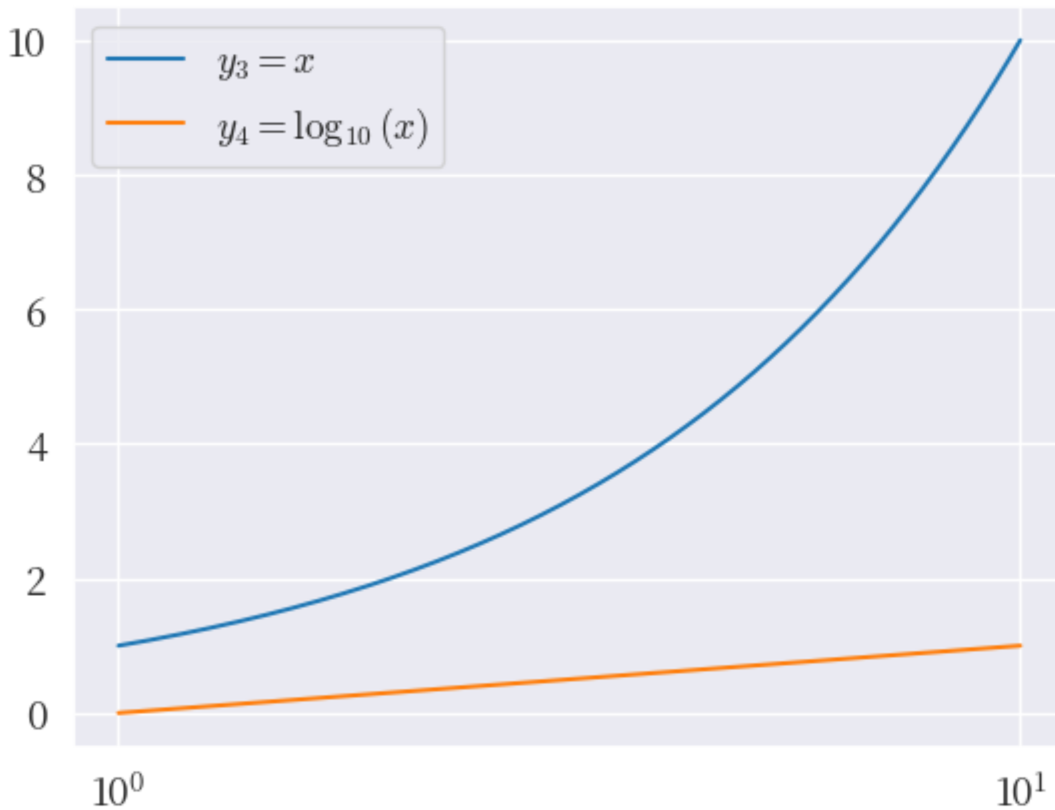
In [ ]:
```
off1 = 4
off2 = 6
sigma = 1.8
x = np.linspace(1, 10, 1000)
y1 = np.exp(-(x-off1)**2/sigma)
y2 = np.exp(-(x-off2)**2/sigma)
y = y1 + y2
plt.plot(x, y1, label="$y_1$")
plt.plot(x, y2, label="$y_2$")
plt.plot(x, y, label="$y_1+y_2$")
plt.legend()
plt.show()
```



对数坐标变换保持形状

正常坐标下的函数f(x)变换到半对数坐标下，只需要绘制f(log(x))即可保持线形不变。

In [ ]:
```
y3 = x
y4 = np.log10(x)
plt.plot(x, y3, label="$y_3 = x$")
plt.plot(x, y4, label="$y_4 = \log_{10}(x)$")
plt.xscale('log')
plt.legend()
plt.show()
```



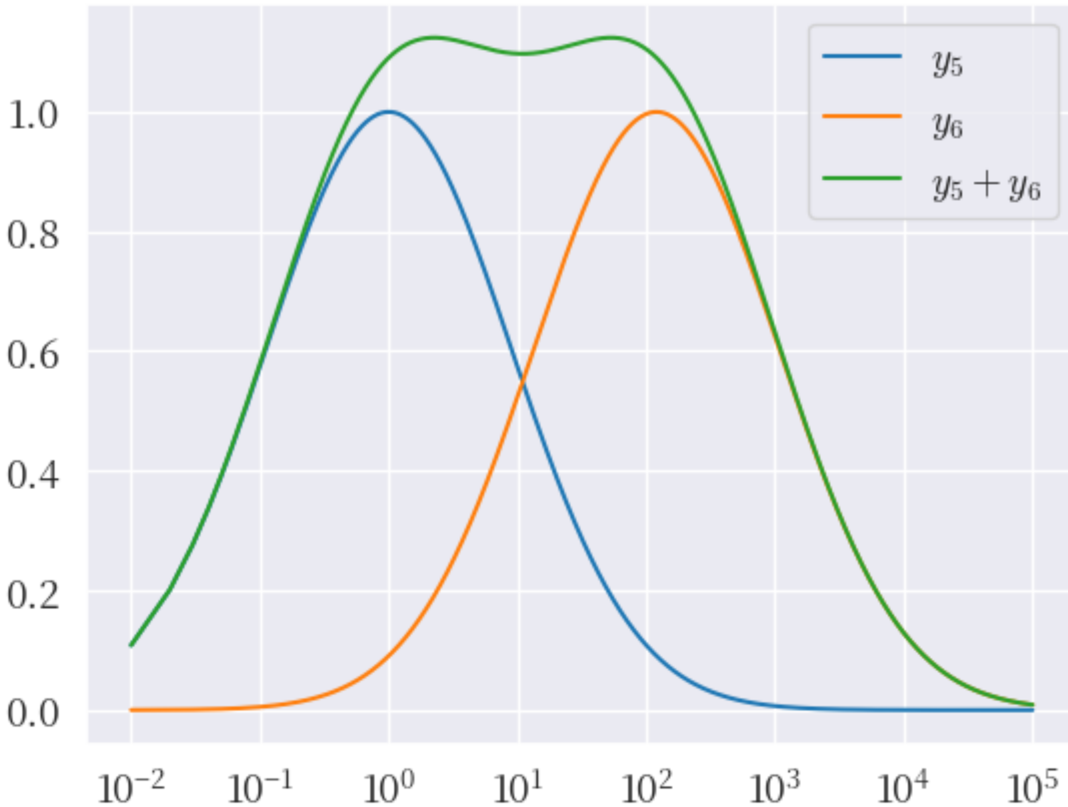
对数下的高斯函数叠加

两个高斯函数的叠加得到类似双峰图像。

In [ ]:
```
c5 = 0
c6 = np.log10(off2*20)
x5 = np.linspace(0.01, 100000, 10000000)
y5 = np.exp(-(np.log10(x5)-c5)**2/sigma)
y6 = np.exp(-(np.log10(x5)-c6)**2/sigma)
suu = (y5 + y6) / x5
print(f'y5 peak at {c5}\ny6 peak at {c6:.3f}')
```

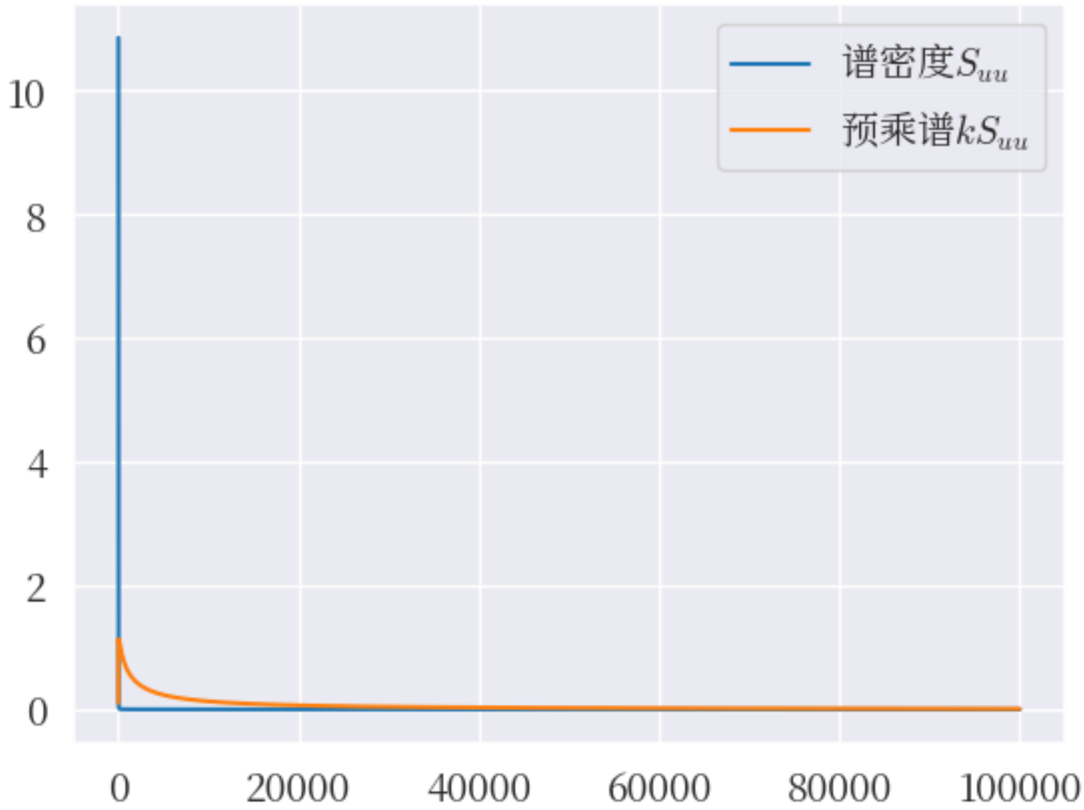
```
y5 peak at 0
y6 peak at 2.079
```

In [ ]:
```
plt.plot(x5, y5, label="$y_5$")
plt.plot(x5, y6, label="$y_6$")
plt.plot(x5, y5+y6, label="$y_5 + y_6$")
plt.legend()
plt.xscale('log')
plt.show()
```



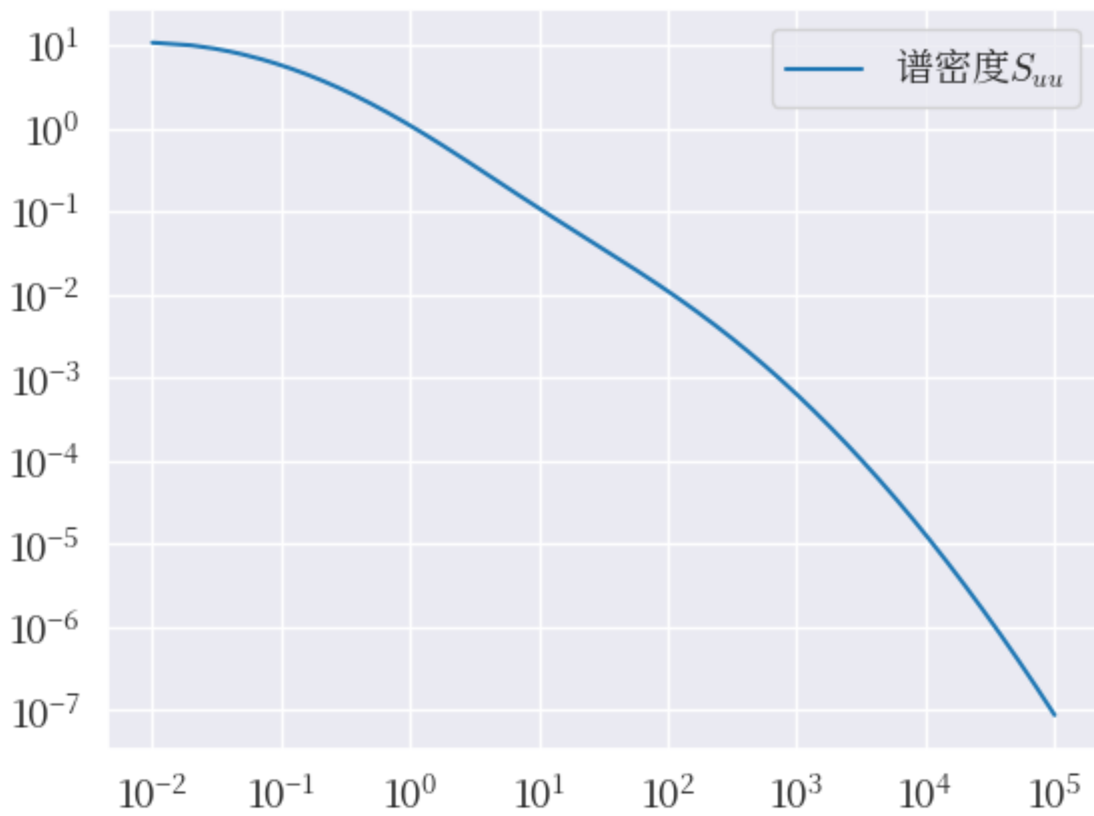
In [ ]:
```
plt.plot(x5, suu, label="谱密度$S_{uu}$")
plt.plot(x5, x5*suu, label="预乘谱$kS_{uu}$")
# plt.xlim(-1, 400)
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7ffa80d7aa50>
```



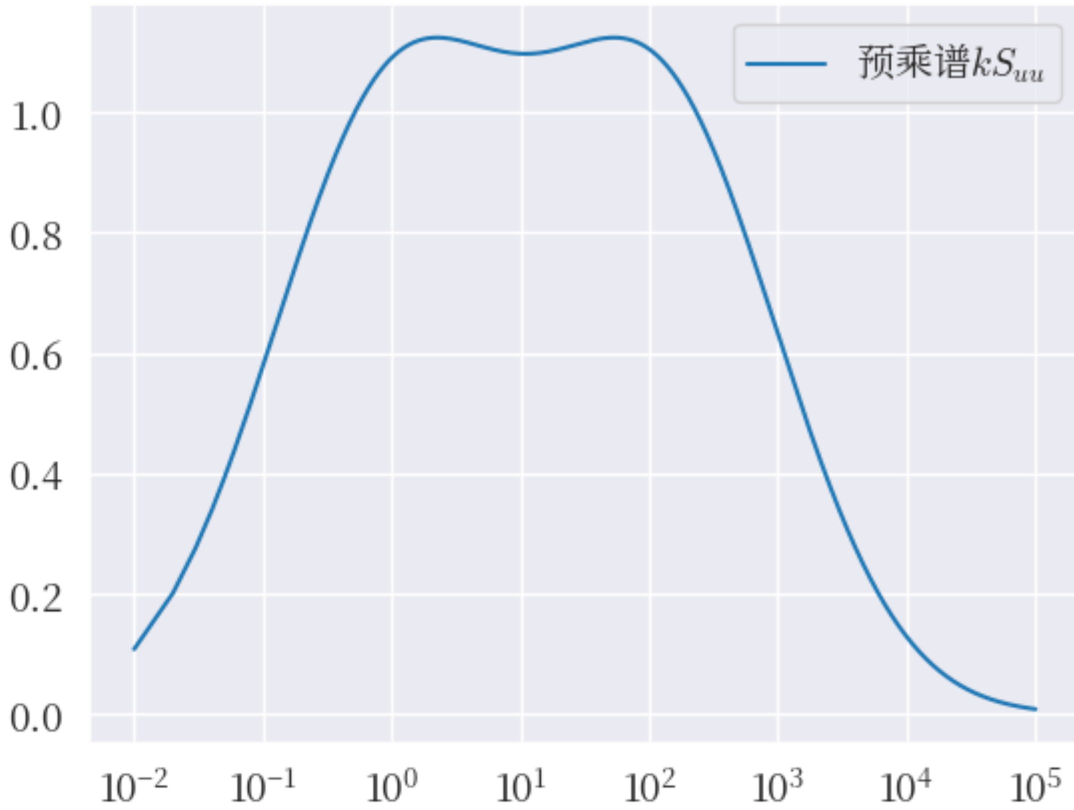
In [ ]:
```
plt.plot(x5, suu, label="谱密度$S_{uu}$")
plt.xscale('log')
plt.yscale('log')
plt.legend()
```

Out[ ]: `<matplotlib.legend.Legend at 0x7ffaa25bd390>`



In [ ]:
```
plt.plot(x5, suu*x5, label="预乘谱$kS_{uu}$")
plt.xscale('log')
plt.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7ffaa27a7b10>
```



坐标轴scale的变换

定义正变换函数及其反变换函数及可得到相应的图形变换。

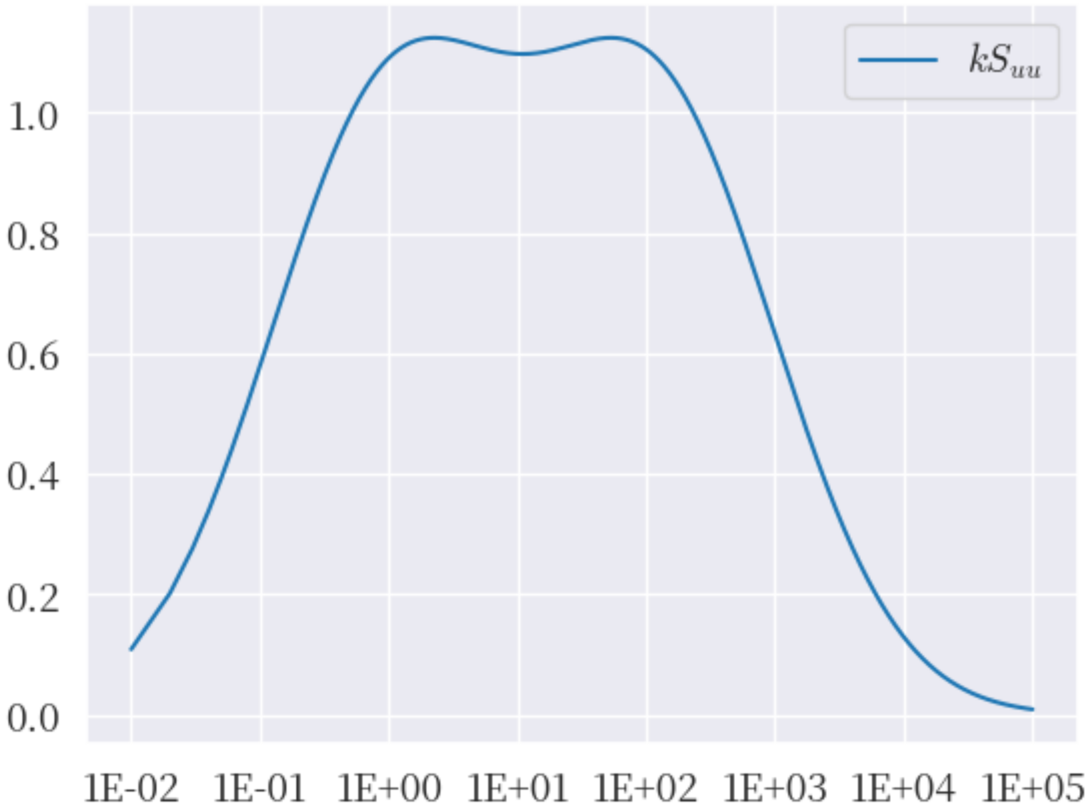
正常坐标下的图形实际上为正反变换函数均为x的trivial case。

In [ ]:
```
def forward(x):
    return np.log10(x)
def inverse(x):
    return np.power(10, x)

fig, ax = plt.subplots(1, 1)
ax.plot(x5, x5*suu, label="$kS_{uu}$")
ax.set_xscale('function', functions=(forward, inverse))
ax.set_xticks([0.01, 0.1, 1, 10, 1e2, 1e3, 1e4, 1e5])
ax.set_xticklabels(["{:.0E}".format(i) for i in [0.01, 0.1, 1, 10, 1e2, 1e3,
ax.legend()
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7ffaa2b75d90>

/tmp/ipykernel_67044/1650559280.py:2: RuntimeWarning: divide by zero encounte
red in log10
  return np.log10(x)
```



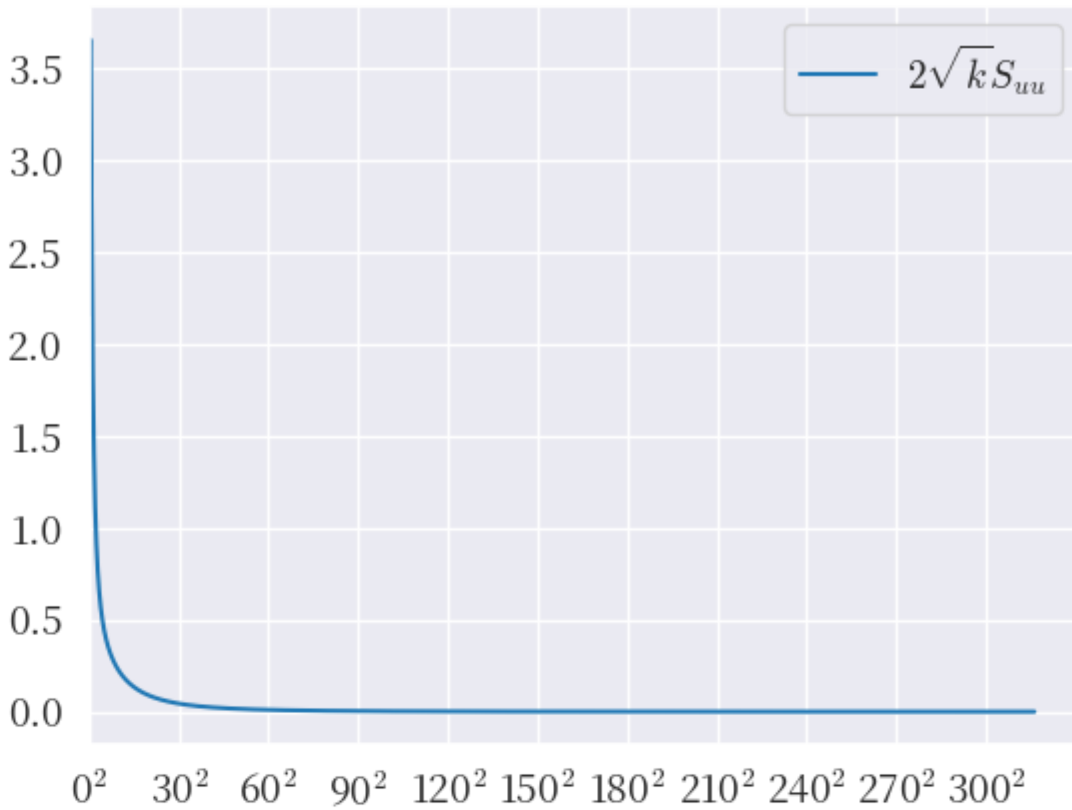
In [ ]:

```python
def forward(x):
    return x**(1/2)
def inverse(x):
    return x**2

fig, ax = plt.subplots(1, 1)
ax.plot(x5, 2*np.sqrt(x5)*suu, label="$2\sqrt{k}S_{uu}$")
ax.set_xscale('function', functions=(forward, inverse))
tiks = [(i*30)**2 for i in range(11)]
ax.set_xticks(tiks)
ax.set_xticklabels(["{:.0f}$^2$".format(np.sqrt(i)) for i in tiks])
ax.legend()
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7ffaa257acd0>
```


$2\sqrt{k}S_{uu}$
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0^2 30^2 60^2 90^2 120^2 150^2 180^2 210^2 240^2 270^2 300^2