In [521…
```
import numpy as np
import math
from scipy.interpolate import make_interp_spline
import matplotlib.pyplot as plt
```

In [522…
```
coin=[1, 5, 10, 25, 50, 100]
cx=np.array(coin)
mean=10.3828
```

In [523…
```
alpha=[-1,0]
t=0
```

In [524…
```
def fn(x):
    ax=x*cx
    g=math.e**ax
    p=g/sum(g)
    px=cx*p
    s=sum(px)
    f=mean-s
    return(f)
```

In [525…
```
while t<100:
    a0=sum(alpha)/2
    if fn(a0)*fn(alpha[0])<0:
        alpha[1]=a0
    else:
        alpha[0]=a0
    t+=1
a0
```

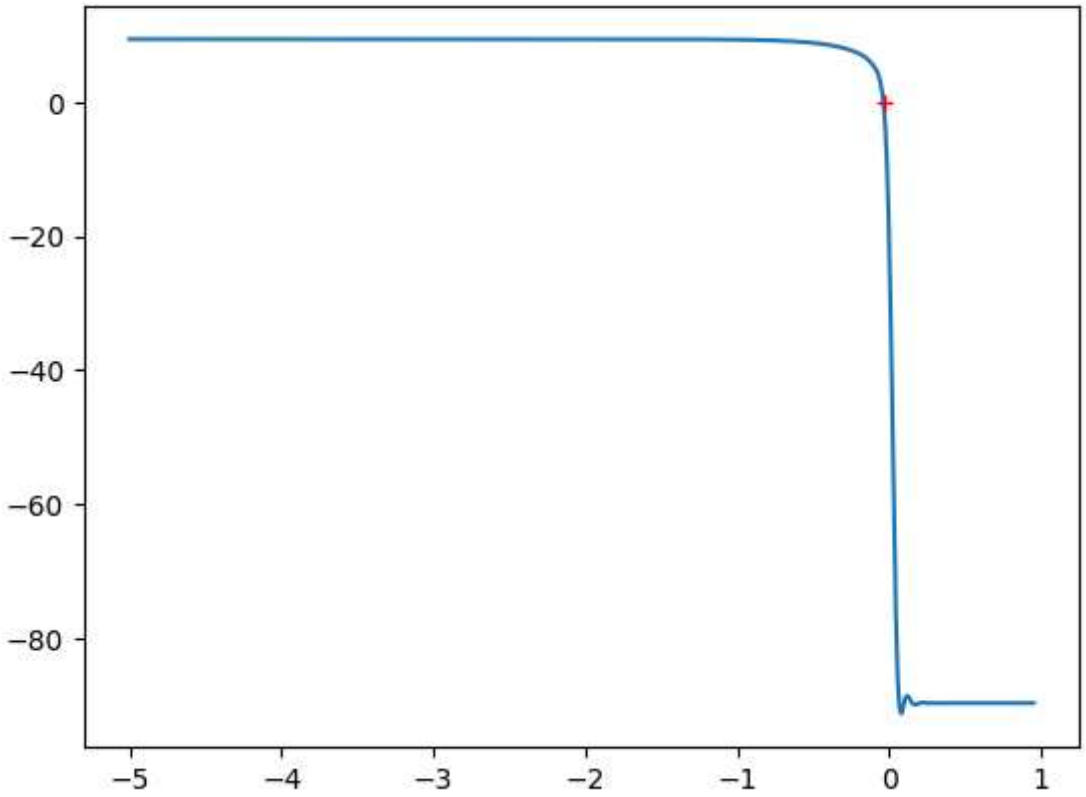
Out[525]:
```
-0.037732955895241005
```

Verification

In [539…
```
x=np.arange(-5,1,0.05)
y=[]
for i in x:
    y.append(fn(i))
```

In [541…
```
X_Y_Spline = make_interp_spline(x, y)
X_ = np.linspace(x.min(), x.max(), 500)
Y_ = X_Y_Spline(X_)
plt.plot(X_, Y_)
plt.plot(a0,fn(a0),'r+')
```

Out[541]:
```
[<matplotlib.lines.Line2D at 0x7f05d7f96410>]
```



In [528…
```
fn(-0.0377)
```

Out[528]:
```
-0.00666488990141012
```

Difference between Estimated and Actual Mean is no larger than .01

In [529…
```
fn(-0.037733)
```

Out[529]:
```
8.913311424052495e-06
```

In [530…
```
fn(-0.037732955895241005)
```

Out[530]:
```
-7.105427357601002e-15
```

Estimated optimal probabilities

In [544…
```
ax=a0*cx
g=math.e**ax
p=g/sum(g)
p
```

Out[544]:
```
array([0.31670281, 0.27233475, 0.22551061, 0.12804386, 0.04985152,
       0.00755645])
```

Estimated Mean

In [545…
```
px=cx*p
s=sum(px)
```

```
s
```

Out[545]: 10.382800000000007

Difference between Estimated and Actual Mean

In [546…

```
mean-s
```

Out[546]: -7.105427357601002e-15