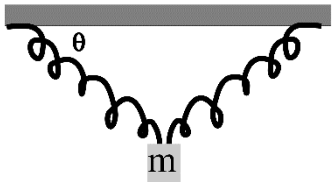

Problem: Find the value of θ at which this system will be equilibrium.

Since we know that the mass of the object is m, the gravity of the object equals to mg. And L_0 is half the distance between the two supports as well as the original length of the spring. So, the change of the spring's length $\Delta L = L - L_0 = \frac{L_0}{\cos(\theta)} - L_0$, and the tension of the spring $T = k\Delta L = k(\frac{L_0}{\cos(\theta)} - L_0)$. Hence, the tension on the vertical direction can be expressed as $T_{\perp} = 2Tsin(\theta) = 2kL_0(\tan(\theta) - \sin(\theta))$ which equals to the gravity of the object. So, the equation which we need to solve is $f(\theta) = \tan(\theta) - \sin(\theta) - \frac{mg}{2kL_0} = 0$.

For this system, the value of θ is limited to $[0, \frac{\pi}{2}]$. Since we've already had an interval, we can firstly use bisection method to get a rough solution, and then "polish it up" by using Newton-Raphson method to get a more accurate solution. In this way, we don't need to search for an appropriate original value for Newton method.

We firstly define the function of bisection method. The function takes 2 parameters as input, the first x_1 refers to the lower bound and the second x_2 refers to the upper bound. In this question, we take 0 as the lower bound, and 1.57 as the upper bound. The reason why we use 1.57 to replace the $\frac{\pi}{2}$ is that if we choose $\theta = \frac{\pi}{2}$, the value of the function $\tan(\theta)$ will tend to infinity which turns out to be a large negative number in program. Inside the function, we define $x_3 = \frac{x_1+x_2}{2}$. And then, we use a while-loop, the while-loop continues when $f(x_3) > 1 \times 10^{-4}$. Inside the while-loop, we use an if-else structure: if $f(x_1) * f(x_3) < 0$ (which means the root is in $[x_1, x_3]$), then $x_2 \leftarrow x_3, x_3 \leftarrow \frac{x_1+x_2}{2}$; else (which means the root is in $[x_3, x_2]$) $x_1 \leftarrow x_3, x_3 \leftarrow \frac{x_1+x_2}{2}$. This function will finally return a rough solution x of the equation, which satisfies $f(x) < 1 \times 10^{-4}$.

Then we define the function of Newton method. The method takes 1 parameter x as input, which is the original value of the iteration. In this question, we can use the rough solution returned from the bisection method as the input. Inside the function, we use a while-loop. We define $a = f(x), b = f'(x)$. If $|a| > 1 \times 10^{-14}$, $x \leftarrow x - \frac{a}{b}$; else ($|a| \leq 1 \times 10^{-14}$) return x and break the while-loop. In this way, we get an accurate solution of the equation.

Input: $f(\theta) = \tan(\theta) - \sin(\theta) - \frac{mg}{2kL_0}$, lower bound x_1, upper bound x_2

Output: The solution x of the equation $f(\theta) = \tan(\theta) - \sin(\theta) - \frac{mg}{2kL_0} = 0$

1. $x_3 \leftarrow \frac{x_1 + x_2}{2}$
2. **While** $f(x_3) > 1 \times 10^{-4}$ **Do**
3.   **If** $f(x_1) * f(x_3) < 0$ **Then**
4. $x_2 \leftarrow x_3, x_3 \leftarrow \frac{x_1 + x_2}{2}$
5.   **Else**
6. $x_1 \leftarrow x_3, x_3 \leftarrow \frac{x_1 + x_2}{2}$
7. **Return** x_3
8. $x \leftarrow x_3$
9. **While** True **Do**
10. $a \leftarrow f(x)$
11. $b \leftarrow f'(x)$
12.   **If** $|a| > 1 \times 10^{-14}$ **Then**
13. $x \leftarrow x - \frac{a}{b}$
14.   **Else**
15.   **Return** x
16.   **Break**

2_3_17307110134.py ×

C: > Users > 11765 > Desktop > 学习 > 物理 > 计算物理 > 作业 > homework2 > 2_3_17307110134.py > Bisection

```python
import sympy as sp
import math as ma

def main():
    root1 = Bisection(0,1.57)
    # First, we use Bisection Method to search for a rough solution.
    # The value of theta satisfies 0° < theta < 90°, so the interval is [0,pi/2].

    root2 = Newton(root1)
    # Then we use Newton Method to polish it up.

    theta = (root2/ma.pi) * 180
    print('The system is in equilibrium at %s°.' % theta)


def Bisection(x1,x2):
# Bisection Method
    root = 0

    if f(x1)*f(x2) < 0:
    # The root is between [x1,x2].
        x3 = (x1+x2) / 2

        while abs(f(x3)) > 1e-4:
            if f(x1)*f(x3) < 0:
```

终端 问题 输出 调试控制台

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\11765\Desktop\学习\物理\计算物理\作业\homework2>  & 'python' 'c:\Users\11765\.vscod
launcher' '56267' '--' 'c:\Users\11765\Desktop\学习\物理\计算物理\作业\homework2\2_3_1730711013
Roughly, 0.5328 is a root of the equation.
f(x)=0.00006476269707, f'(x)=0.48631455376744, the times of iteration is 0.
f(x)=0.00000001859504, f'(x)=0.48603530600667, the times of iteration is 1.
f(x)=0.00000000000000, f'(x)=0.48603522777557, the times of iteration is 2.
0.53265487729242 is a root of the equation.
The system is in equilibrium at 30.5188764059145°.
PS C:\Users\11765\Desktop\学习\物理\计算物理\作业\homework2>
```

The picture shows the result after running the program.

By using the bisection method, the program shows that "Roughly, 0.5328 is a root of the equation."

Then by using the Newton method, after 2 times of iteration, the program gives out a more accurate root. After changing the rad into degree, it finally shows that "The system is in equilibrium at 30.5188764059145°."