

计算物理第五次 Ex

白博臣 2022141220036

Problem 1

使用二分法进行求解方程 $2 - x - e^{-x}$ 的根，构建一个迭代关系与条件判断，代码如下：

```
1 import numpy as np
2
3
4 def f(x):
5     return 2 - x - np.exp(-x)
6
7
8 def bisection_method(f, a, b, tol=1e-6, max_iter=100):
9     if f(a) * f(b) >= 0:
10         print("Bisection method fails.")
11         return None
12
13     iter = 0
14     while (b - a) / 2 > tol and iter < max_iter:
15         c = (a + b) / 2
16         if f(c) == 0:
17             break
18         if f(a) * f(c) < 0:
19             b = c
20         else:
21             a = c
22         iter += 1
23
24     return c
25
26
27 # 设置初始区间[a, b]
28 a = -2
29 b = 0
30 root_1 = bisection_method(f, a, b)
31 a = 1
32 b = 3
33 root_2 = bisection_method(f, a, b)
34 if root_1 is not None:
```

```
35 print(f"Root found at x = 第一个根: {root_1},第二个根: {root_2}")
```

由于可以肉眼观察出该方程有两个解,通过大致猜测的方法确定出两个根的范围分别进行求解即可。

输出结果:

```
Root found at x = 第一个根: -1.1461925506591797,第二个根: 1.8414058685302734

进程已结束,退出代码为 0
```

Problem 2

采用割线法,只需基于 Problem 1 的代码按照割线法的迭代关系进行修改即可,代码如下:

```
1 import numpy as np
2
3
4 def secant_method(f, a, b, tol=1e-6, max_iter=100):
5     """
6
7     :param f: 所需求解的函数
8     :param a: a 是指的  $p_{(n-1)}$ 
9     :param b: b 是指的  $p_{(n)}$  c 指的是  $p_{(n+1)}$ 
10    :param tol: 最小精度
11    :param max_iter: 最大迭代次数
12    :return:
13    """
14    iter = 0
15    while abs(b - a) > tol and iter < max_iter:
16        c = b - f(b) * (b - a) / (f(b) - f(a))
17        a, b = b, c
18        iter += 1
19
20    return c
21
22
23 # 定义要求解根的函数
24 def f(x):
25     return 2 - x - np.exp(-x)
26
27
28 # 设置初始猜测值 a 和 b
```

```
29 a = -2
30 b = 0
31
32 root_1 = secant_method(f, a, b)
33
34 a = 0
35 b = 3
36 root_2 = secant_method(f, a, b)
37 print(f"Root found at x = 第一个根: {root_1},第二个根: {root_2}")
```

输出结果如下:

```
Root found at x = 第一个根: -1.1461932206205816,第二个根: 1.8414056604369609

进程已结束,退出代码为 0
```