

**运筹学**

**优化算法编程作业**

姓 名： 刘博文

学 号： M202221325

班 级： 协创硕2203

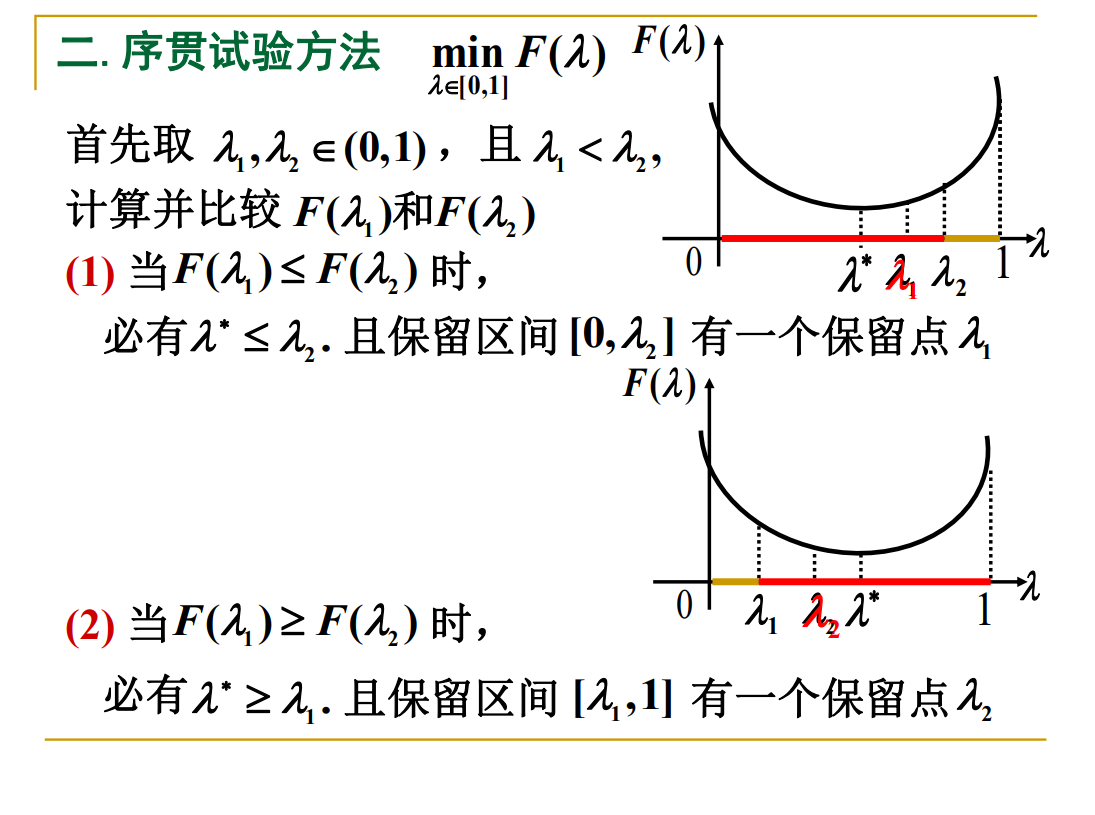
任课教师： 赵金玲

2022年 10 月26日

1. **问题描述及算法原理**

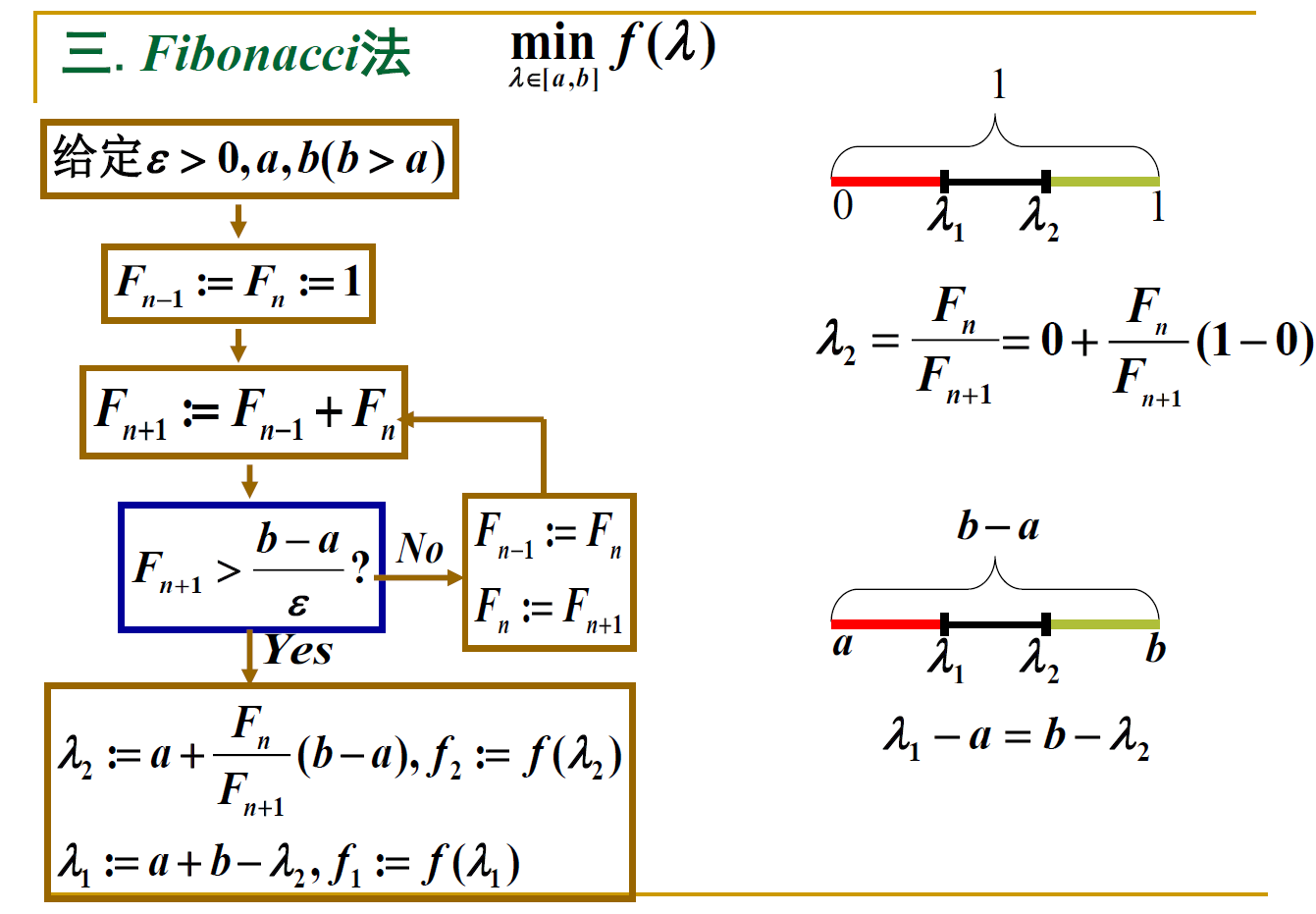
一维搜索算法是求解一元函数的无约束极值问题的数值解法。它是求解元函数的无约束极值问题的重要组成部分。

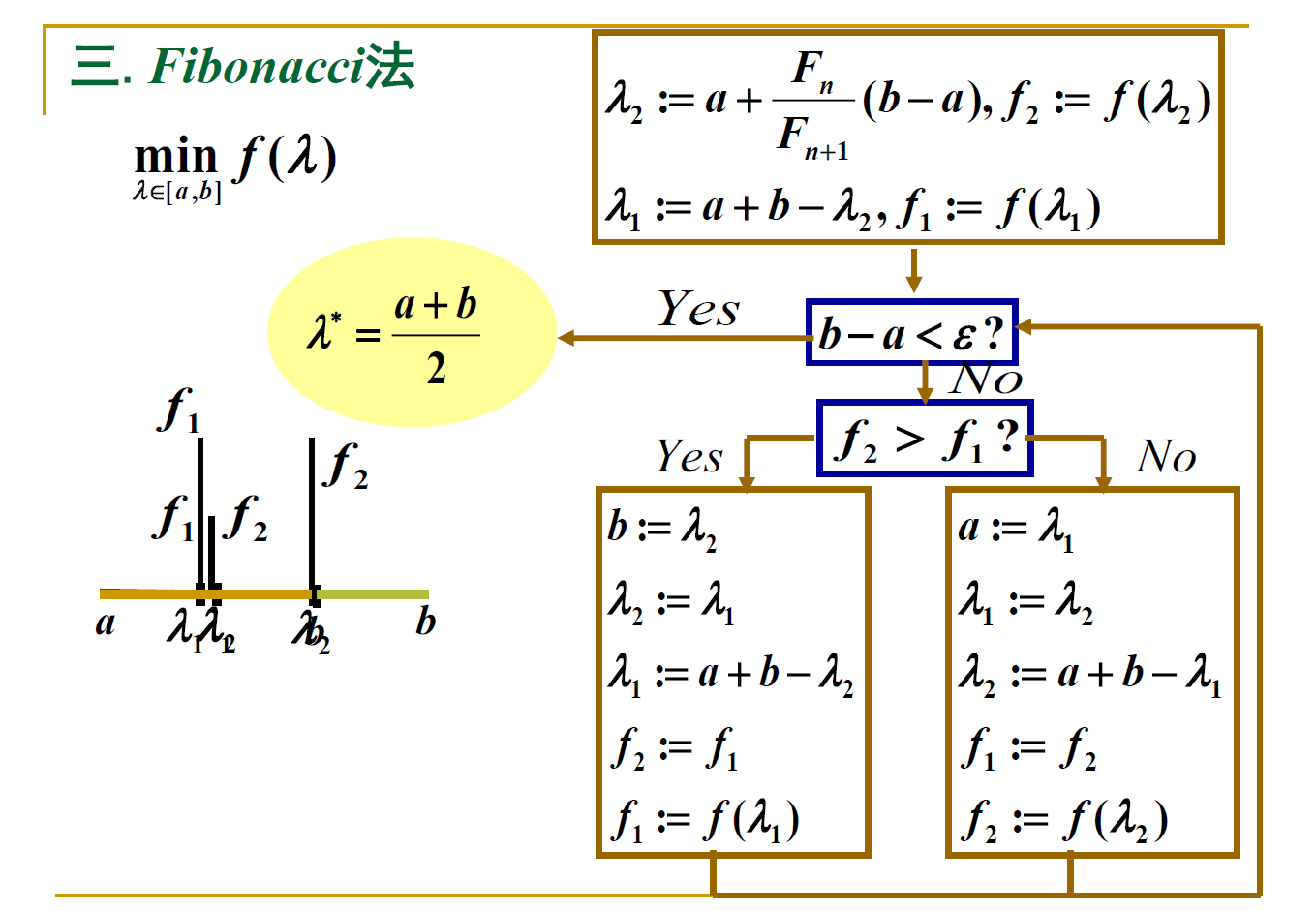
在一维搜索算法中，序贯试验方法是一种重要的求解手段，关于这个方法可通过举例说明，假设函数在[0,1]上有唯一的极小点，首先取，且，计算并比较和，此时会出现两种情况，如下图所示：



其中，序贯试验方法中所包含的Fibonacci法和0.618法是两个常用的一维搜索求解算法，本实验所实现的即Fibonacci法和0.618法。

首先来看Fibonacci法，它的求解步骤可以由下图所示：





0.618法与Fibonacci法的区别仅在于初始点的选取不一样，在Fibonacci法中，初始点分别为。而0.618法为.

1. **编程实现**

1) Fibonacci法

1. '''''
2. Author: 刘博文
3. Date: 2022-10-14 13:28:40
4. LastEditTime: 2022-10-14 16:11:46
5. Description: 无约束最优化问题————Fibonacci法
6. https://github.com/touchfisher
7. Copyright (c) 2022 by touchfisher 1632570150@qq.com, All Rights Reserved.
8. '''
9. **def** f(x):
10. **return** x\*\*2 + 2\*x
12. **if** \_\_name\_\_ == '\_\_main\_\_':
13. e,a,b = map(float,input("请输入精度(如0.02),以及区间的左右端点a和b,全部以空格隔开:").split())
14. Fibonacci = [1,1]
15. Fibonacci\_next = Fibonacci[0] + Fibonacci[1]
16. **while**(Fibonacci\_next < (b-a)/e):
17. Fibonacci[0] = Fibonacci[1]
18. Fibonacci[1] = Fibonacci\_next
19. Fibonacci\_next = Fibonacci[0] + Fibonacci[1]
20. lambda2 = a + Fibonacci[1] / Fibonacci\_next \* (b-a)
21. f2 = f(lambda2)
22. lambda1 = a + b - lambda2
23. f1 = f(lambda1)
24. **while**(b - a > e):
25. **if** f2 > f1:
26. b = lambda2
27. lambda2 = lambda1
28. lambda1 = a + b - lambda2
29. f2 = f1
30. f1 = f(lambda1)
31. **else**:
32. a = lambda1
33. lambda1 = lambda2
34. lambda2 = a + b - lambda1
35. f1 = f2
36. f2 = f(lambda2)
37. # 最优点默认保留6位小数，最优值默认保留到整数，如需显示更高精度可去掉round函数
38. lambda\_finally = round((a + b) / 2, 6)
39. **print**(f"最优点为{lambda\_finally},最优值为{round(f(lambda\_finally))}")

2) 0.618法

1. '''''
2. Author: 刘博文
3. Date: 2022-10-14 15:11:12
4. LastEditTime: 2022-10-26 20:01:36
5. Description: 无约束最优化问题————0.618法
6. https://github.com/touchfisher
7. Copyright (c) 2022 by touchfisher 1632570150@qq.com, All Rights Reserved.
8. '''
9. **def** f(x):
10. **return** x\*\*2 + 2\*x
12. **if** \_\_name\_\_ == '\_\_main\_\_':
13. e,a,b = map(float,input("请输入精度(如0.02),以及区间的左右端点a和b,全部以空格隔开:").split())
14. lambda2 = a + 0.618 \* (b-a)
15. f2 = f(lambda2)
16. lambda1 = a + b - lambda2
17. f1 = f(lambda1)
18. **while**(b - a > e):
19. **if** f2 > f1:
20. b = lambda2
21. lambda2 = lambda1
22. lambda1 = a + b - lambda2
23. f2 = f1
24. f1 = f(lambda1)
25. **else**:
26. a = lambda1
27. lambda1 = lambda2
28. lambda2 = a + b - lambda1
29. f1 = f2
30. f2 = f(lambda2)
31. # 最优点默认保留6位小数，最优值默认保留到整数，如需显示更高精度可去掉round函数
32. lambda\_finally = round((a + b) / 2, 6)
33. **print**(f"最优点为{lambda\_finally},最优值为{round(f(lambda\_finally))}")
34. **结果测试**

本实验