

MA348 Numerical Analysis, Optimization of Functions

David Jefts

March 28th, 2019

Introduction

This lab is based off of using The Golden Ratio (φ) as a constant proportional value to find the extremum (maximum or minimum) of a function. The goal of this lab is to use the Golden-Section Search bracketing method to find the maximum of the function below:

$$f(x) = 2\sin(x) - x^2/10 \quad (1)$$

Theory-Analysis

The Golden Ratio Search method is a bracketing method that involves recursively modifying the upper and lower bounds that the function searches on until the two bounds are equivalent within the specified tolerance. The bound to be changed is the bound that furthest from the local vertex of the equation. From hereon-in, the left-bound will be referred to as a , and the right-bound is b . The ‘new’ values for both of these bounds are a_1 and b_1 , respectively. This search method seeks to change each bound proportionally, as opposed to a set value, to prevent ‘overshooting’ the target vertex value. The 3 initial equations that represent this are:

$$(b_1 - a) = s(b - a) \quad (2)$$

$$(b - a_1) = s(b - a) \quad (3)$$

$$(a_1 - a) = s(b_1 - a) \quad (4)$$

Where s is some (currently unknown) numerical constant, Equation 2 checks that the new b value is proportional to the original range, Equation 3 confirms that the new a value is proportional to the original range, and Equation 4 certifies that the change in a is proportional to the new range of values.

Substitute (2) into (4):

$$(a_1 - a) = s^2(b - a) \quad (5)$$

Add (3) to (5) and simplify:

$$(b - a) = s^2(b - a) + s(b - a) \quad (6)$$

Divide (6) by $(b - a)$:

$$1 = s^2 + s \quad (7)$$

Rearrange (7) into a quadratic and then solve for s :

$$0 = s^2 + s - 1 \quad (8)$$

$$s = -\frac{1 \pm \sqrt{5}}{2} \quad \text{or} \quad s = 0.618, -1.618$$

For this lab, only the positive root was necessary and was estimated to be $s \approx 0.6180339887$.

To calculate the new a or b value, solve (2) or (3) for b_1 or a_1 , respectively:

$$b_1 = a + s(b - a) \quad (9)$$

$$a_1 = b - s(b - a) \quad (10)$$

Numerical Solution

The entirety of the Golden-Section Search maximum search was coded in Python, with the report created in L^AT_EX.

This method is very similar to other bracketing methods such as The Bisection Method or The Newton-Raphson Method, except with it the goal is to calculate the extrema of a function instead of the roots. It requires two initial guesses, one on either side of the supposed location of the extrema. For this lab, the initial guess was $[0, 4]$ where the left value is represented by the variable a and is the lower bound of the search space, while the right value is represented by the variable b and is the upper bound of the search space.

The Golden-Section Search is an iterative method, continuing until $(b-a)$ is less than a predetermined tolerance, 1×10^{-5} was used during testing. Each iteration calculates the values for a_1 and b_1 , and compares the result of substituting them into the base function. When trying to find the maximum, if $f(a_1) \geq f(b_1)$, then the b value is replaced by b_1 . If not, then the a value is replaced by a_1 . This moves the edge of the range furthest from the maximum towards the maximum value.

One drawback to this method is that it requires a slightly different algorithm when trying to find the minimum of a function: if $f(a_1) \leq f(b_1)$, then the b value is replaced by b_1 . If not, then the a value is replaced by a_1 . This is because the comparison uses an inequality to determine the bounds of the next iteration.

Results and Discussion

The table of the a and b values as described in the Numerical Solution Section is shown below:

$\mathbf{a_1}$	$\mathbf{b_1}$
0	4
0	2.472135955
0.94427191	2.472135955
0.94427191	1.88854382
1.304951685	1.88854382
1.304951685	1.66563146
1.304951685	1.527864045
1.39009663	1.527864045
1.39009663	1.475241575
1.39009663	1.4427191
1.410196625	1.4427191
1.422619105	1.4427191
1.422619105	1.435041585
1.422619105	1.43029662
1.425551655	1.43029662
1.425551655	1.428484204
1.426671789	1.428484204
1.426671789	1.427791923
1.427099642	1.427791923
1.42736407	1.427791923
1.42736407	1.427628498
1.427465073	1.427628498
1.427527496	1.427628498
1.427527496	1.427589918
1.427527496	1.427566075
1.427542232	1.427566075
1.427542232	1.427556968

The maximum value for the function $f(x) = 2\sin(x) - x^2/10$ is approximately 1.7757256531306822 when $x \approx 1.42754959962$.

Conclusions

The Golden-Section Search method is very effective at finding the extrema of functions, however it seems to take an exceptional amount of iterations to achieve small tolerances. Additionally, the method requires calculating a_1 and b_1 every iteration which is inefficient. In conclusion, the Golden-Section Search Method is very effective at finding extrema and guarantees a solution, but it requires different methods for minimum and maximum and does not seem particularly efficient.

Appendix A

```
27 class GoldenSearch:
28     def __init__(self, tol):
29         self.tol = tol
30         self.s = 0.6180339887
31
32     def min(self, a, b):
33         err = abs(b - a)
34         a1 = None
35         while err > self.tol:
36             b1 = a + self.s * (b - a)
37             a1 = b - self.s * (b - a)
38             print(a, b)
39             if f(a1) < f(b1):
40                 a = a1
41             else:
42                 b = b1
43             err = abs(b - a)
44         return f(a1)
45
46     def max(self, a, b):...
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61 def f(x):
62     # return x**2 + 2 * x # find min
63     # return -x**2 + 2 * x # find max
64     return 2 * math.sin(x) - (x**2 / 10) # find max for lab #6
65
66
67 tolerance = 0.00001
68 g1 = 0 # starting interval
69 g2 = 4 # ending interval
70 gold = GoldenSearch(tolerance)
71 print("Solution:", gold.max(g1, g2))
```