

Branch: master ▾

Find file

Copy path

ENMT-4620 / Week 3 / Writeup_Golden_Section.md

 ejbkdb Create Writeup_Golden_Section.md

c79e455 20 hours ago

1 contributor

Raw Blame History



74 lines (56 sloc) 2.57 KB

Golden Section Method Writeup

This writeup with step through the code used to create a sequential quadratic approximation for minimizing an equation

Complete code can be found here [link to code](#):

Equation Definition:

- to utilize simply pass the x-variable to the function. Function will return y value.

```
def obj(x):  
    return 5 - x + 0.45 * x**2 - 0.08**3 + 0.005*x**4
```

Initial Setup:

- To utilize the golden section method you need to define starting points, the starting points need to reside on each side of the curve you are trying to find the minimum for. Startpoints (x_up, x_lo)
- Initialize golden ratio value, and find "d". d is used for defining the next x_up and x_lo points in my case called(new_x_up, new_x_lo). In other literature they are defined x1,x2
- Initialize function for tracking x_up and x_lo as the program iterates. I use XLO and XHI to log these values.

```
x_up = 20  
x_lo = -20
```

```
golden_ratio = ((np.sqrt(5)-1)/2)  
d = golden_ratio*(x_up-x_lo)  
new_x_up = x_lo + d  
new_x_lo = x_up - d
```

```
XLO = []  
XHI = []
```

Generating new "x" values

- These conditional statements determine which value gets update x_up or x_lo. Ultimately this allows x_up and x_lo to iterate idendently based on how close they are to the optimum.

```
if func(new_x_lo) < func(new_x_up):  
    x_up = new_x_up  
    new_x_up = new_x_lo  
    d = golden_ratio*(x_up-x_lo)  
    new_x_lo = x_up - d
```

```
XLO.insert(0,new_x_lo)
if func(new_x_lo) > func(new_x_up):
    x_lo= new_x_lo
    new_x_lo = new_x_up
    d = golden_ratio*(x_up - x_lo)
    new_x_up = x_lo + d
XHI.insert(0,new_x_up)
```

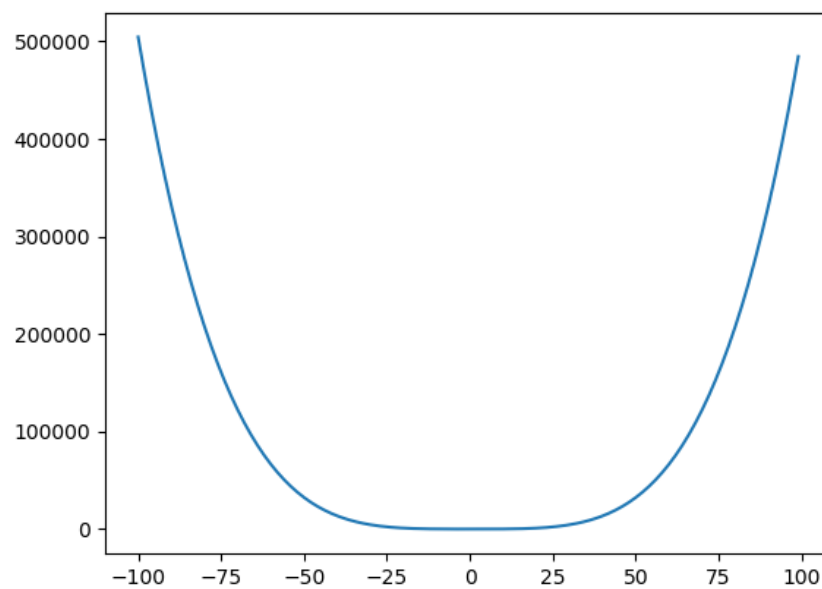
Exit Condition:

- Makes sure XLO and XHI have iterated at least once.
- Then if the difference between iterations of XLO and XHI are < 0.1 the program exits

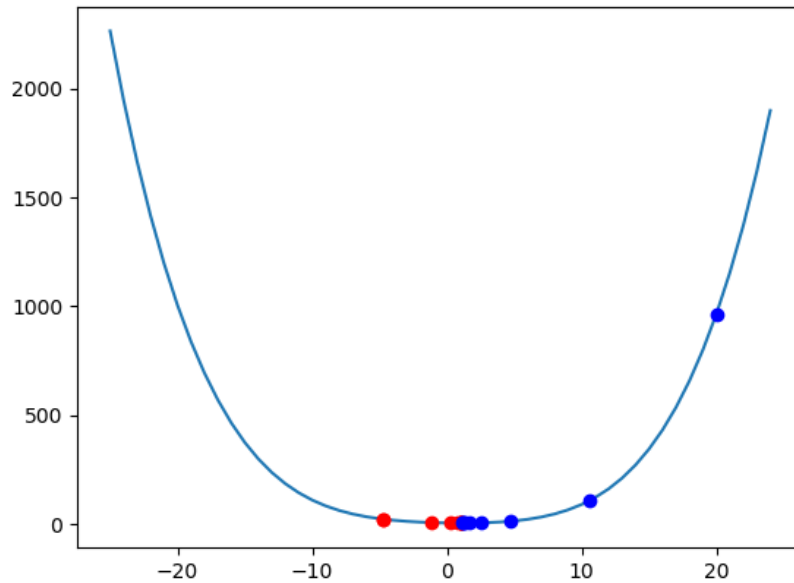
```
if len(XLO) and len(XHI) > 1:
    if abs(XLO[0] - XLO[1]) and abs(XHI[0] - XHI[1]) < 0.1:
        break
```

Results:

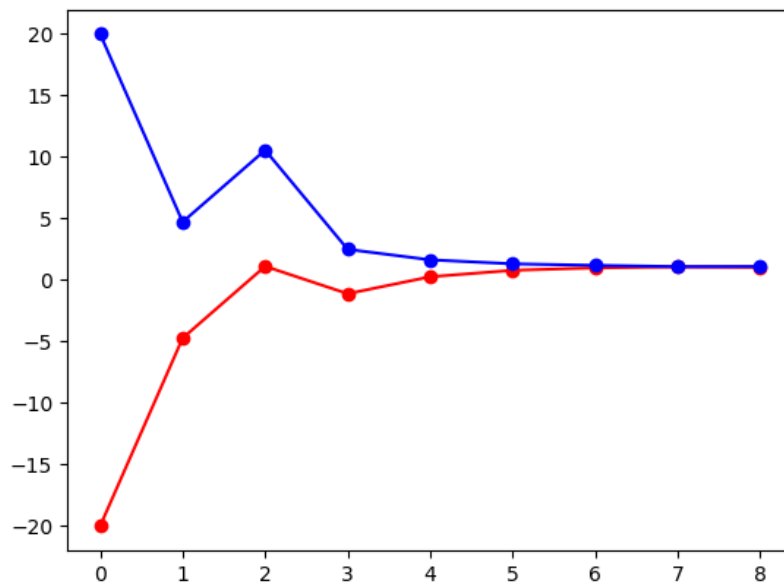
- Graph OF vs. DV



- OF vs. Iterations of Golden Section



- Golden Section Convergence



Result Check:

Utilized `scipy.optimize.minimize_scalar(OF)` results = 1.08, Golden section result = 1.08. Results checked out!