Applied Computational Methods in Mechanical Sciences

(ME466)

Assignment 5

Himanshu Kumar

16ME234

August 30, 2019

Problem Statement:

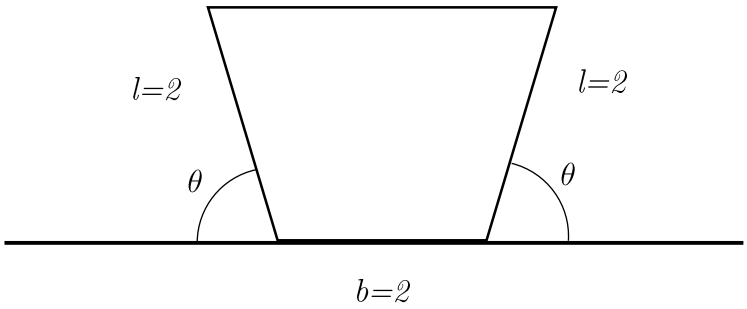
Consider the figure given below. The cross sectional area A of a gutter with base length b=2 and edge length l=2 is given by:

$$A = 4sin\theta(1 + cos\theta)$$

Find the angle which maximizes the cross-sectional area. Use Golden-section method with:

$$\theta \in [0, \frac{\pi}{2}]$$

and compare the results when Newton's method is used with and initial guess as: $\theta = {}^{\pi}/_{4}$.



Relative error limit is taken as 0.00000001.

Python Code:

```
import math
import time

def area(y,theta):
    return(4*math.sin(theta)*(1+math.cos(theta)))
```

```
def golden_sec(f,args,range_):
    phi = (pow(5,0.5)-1)/2
    a = range_[0]
    b = range_[1]
    err_lim = 0.00000001
    itr = 0

    while(1):
        itr = itr+1
        d = phi*(b-a)
        x1 = a+d
        x2 = b-d

        arg_v1 = args +(x1,)
        v1 = f(*arg_v1)

        arg_v2 = args +(x2,)
        v2 = f(*arg_v2)

        if(v1>v2):
            a = x2
        elif(v2>v1):
            b = x1
        else:
            res = a
            break

        err = b-a
        if(err<0):
            err = (-1)*err

        res = (b+a)/2
        # print("\n\nval is :",res)
        # print("error is :",err)

        if(err<err_lim):
            break
```

```
    print("\n\nresult is ",res)
    print("\n Iterations:",itr)
    print ("\n CPU time: ", time.process_time(),'s')
    return(res)

def f_d(x):
    return( 4*( math.cos(x) + math.cos(2*x) ) )
def f_dd(x):
    return( -4*( math.sin(x) + 2*math.sin(2*x) ) )

def newton(f_d,f_dd,init_guess):
    err_lim = 0.00000001
    itr = 0
    x = init_guess

    while(1):
        itr = itr+1
        y = x - (f_d(x)/f_dd(x))

        err = (y-x)/y
        if(err<0):
            err = (-1)*err
        # print("\n\nval is :",y)
        # print("error is :",err)
        if(err<err_lim):
            break
        x=y

    print("\n\nresult is ",y)
    print("\n Iterations:",itr)
    print ("\n CPU time: ", time.process_time(),'s')
    return(y)

#ans = golden_sec(area,(3,),(0,(math.pi/2)))
ans = newton(f_d,f_dd,(math.pi/4))
```

Results:

1. **With golden-section method:**

   ```
   result is  1.0471975355130767

   Iterations: 38

   CPU time:  0.40625 s
   ```

2. **With Newton's method:**

   ```
   result is  1.0471975511965976

   Iterations: 4

   CPU time:  0.1875 s
   ```