Printout

Monday, September 20, 2021 1:41 PM

Section 1

MANE 3351

Subsection 1

Lecture 9, September 20

Classroom Management

Agenda

- Example problem from Wednesday
- Lecture
- Lab today!

Subsection 2

Resources

Handouts

- Lecture 9 Slides
- Lecture 9 Marked Slides

Calendar

Lecture	Date	Content
1	August 23	Welcome to Class, Syllabus
2	August 25	Error Analysis
3	August 30	Introduction to Jupyter Notebook
4	September 1	Markdown
5	September 6	Labor Day Holiday
6	September 8	Taylor Polynomials, Homework 1
7	September 13	Roots of Equations, Bisection Method
8	September 15	Bisection Error Analysis
9	September 20	False Position Algorithm

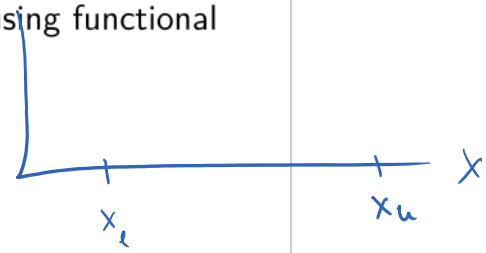
Assignments

- Homework 2 (assigned 9/15/2021, due 9/23/2021)

False Position

- Attempts to converge faster than bisection method by using functional information
- Referred to as the secant method in textbook
- The new root is found by

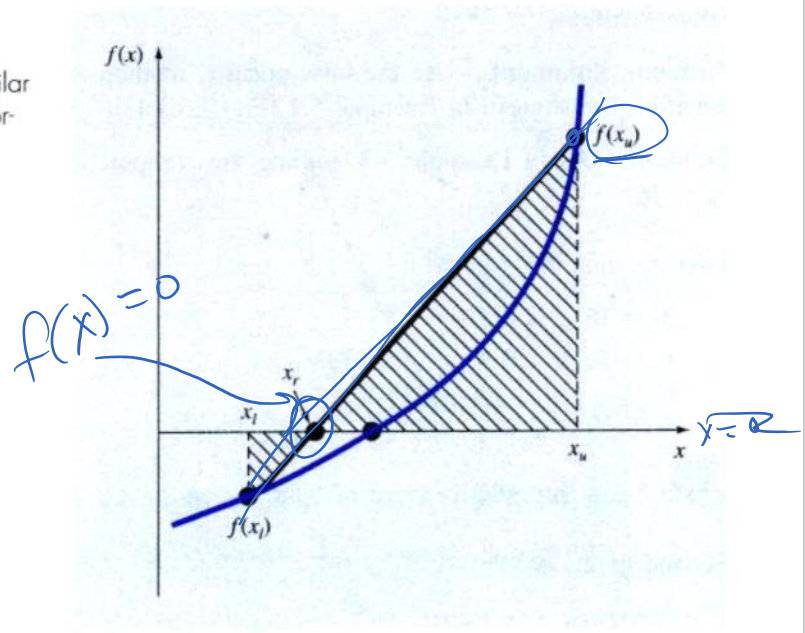
$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$



$\frac{\Delta x}{\Delta y} \rightarrow \frac{\Delta y}{\Delta x}$ slope

- Figure 5.12 (Chapra and Canale(2015)[^1] illustrates the false position method

FIGURE 5.12
A graphical depiction of the method of false position. Similar triangles used to derive the formula for the method are shaded.



False Position Code

The pseudocode, shown below was taken from [^2]

Complete Pseudocode for Regula Falsi or False Position Method

```
1. Start

2. Define function f(x)

3. Input
        a. Lower and Upper guesses x0 and x1
        b. tolerable error e

4. If f(x0)*f(x1) > 0
        print "Incorrect initial guesses"
        goto 3
   End If

5. Do

        x2 = x0 - ((x0-x1) * f(x0))/(f(x0) - f(x1))

        If f(x0)*f(x2) < 0
                x1 = x2
        Else
                x0 = x2
        End If

   While abs(f(x2) > e

6. Print root as x2
```

Jupyter Notebook Demonstration

False Position Error Analysis

- Much harder to analyze than bisection method
- Convergence to root depends on shape of $f(x)$
- At times, false position may converge slowly!, Example from textbook[^3], shown below

Figure 2.7.1: A troublesome function for the bracketed secant method.

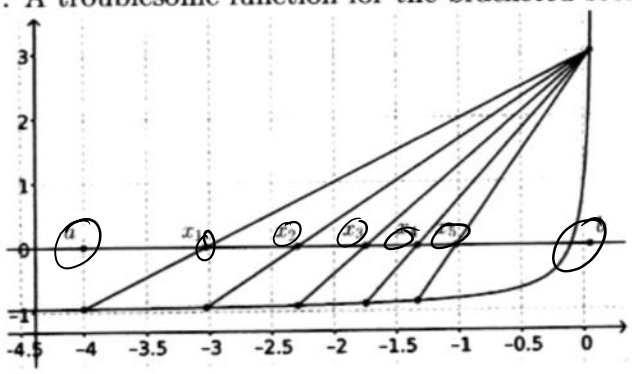


Figure 2: False Position troublesome function

Loops in Python

- Python only supports two types of loops: for and while
- Pseudocode used do/while loop, repeat until

[1]: Chapra, S. and Canale, R. (2015), *Numerical Methods for Engineering, 7th edition*

[3]: Brin, L., (2020), *Tea Time Numerical Analysis: Experiences in Mathematics, 3rd edition*

Monday, September 20, 2021 2:07 PM

Standard normal 1st Quartile

find z s.t. $\Phi(z) = .25$

$f(z) = \Phi(z) - .25 = 0$

Crude bounds

$a = -3.00$ $f(a) > 0$

$b = 0$ $f(b) < 0$

tolerance .001, what is value of n?

$$n > \frac{\log(b-a) - \log 2\varepsilon}{\log 2} = \frac{\log(3) - \log 2(.001)}{\log 2}$$

$= 10.55 \rightarrow 11$ steps

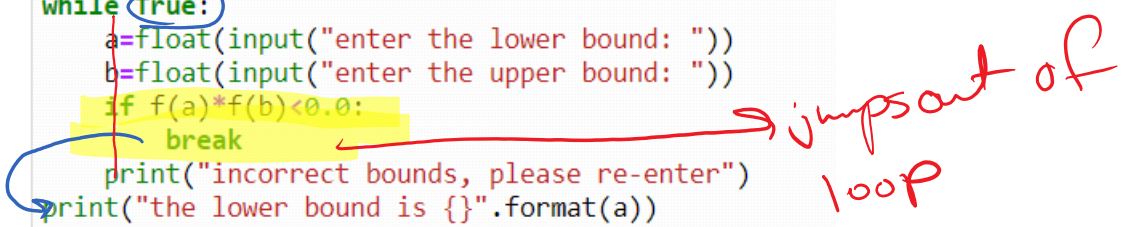
what is error after 3 iterations

$$e_n = \left| \frac{1}{2^{n+1}} (b-a) \right|$$

$$e_3 = \left| \frac{1}{2^{3+1}} (.3-0) \right| = 3/16 = .1875$$

Monday, September 20, 2021 2:21 PM

```
# False
from scipy import stats
import math
def f(x):
    return stats.norm.cdf(x)-.25
# input a & b
while True:
    a=float(input("enter the lower bound: "))
    b=float(input("enter the upper bound: "))
    if f(a)*f(b)<0.0:
        break
    print("incorrect bounds, please re-enter")
print("the lower bound is {}".format(a))
print("the upper bound is {}".format(b))
```



Screen clipping taken: 9/20/2021 2:21 PM

Monday, September 20, 2021 2:25 PM

```
counter=0
# process loop
while True:
    m= a - ((a-b)*f(a))/(f(a)-f(b))
    if f(a)*f(b)<0.0:
        b=m
    else:
        a=m
    counter=counter+1
    if math.fabs(f(m)) <0.0005:
        break
print("the root is {}".format(m))
print("the value at the root is {}".format(f(m)))
print("the number of steps were {}".format(counter))
```

for ( ):

or

while ( ):

Screen clipping taken: 9/20/2021 2:25 PM