Section 1

MANE 3351

Lecture 8

Classroom Management

Agenda

- Part 2: Bisection Search lecture
- Test 1

Resources

Handouts

- Lecture 8 Slides
- Lecture 8 Marked Slides
- Test 1 - Fall 2023

Assignments

- Homework 1 (assigned 9/16/2024, due 9/23/2024 (before 11:59 pm)
- Homework 2 (assigned 9/18/2024, due 9/25/2024 (before 11:59 pm - no late submissions)
- Lab 3 (assigned 9/18/2024, due 9/25/2024 (before 2:00pm))
- Read textbook pages 41-46

Schedule

Lecture/Lab	Date	Topic
8	9/23	Roots of Equations, bisection method (not on Test 1)
9	9/25	Bisection Method Error Analysis, False Position (not o
10	9/30	Test 1 (lectures 1-7)

Roots of Equations

Introduction

- Value x such that $f(x)=0$
- Extremely useful operations
- Engineering Economic Examples
    - Break-even analysis
    - Payback period
    - Rate of return

Quadratic Equation

Consider a second-order polynomial $ax^2 + bx + c = 0$

- The quadratic formula finds the value(s) of x

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Root Bracketing Techniques

- Many techniques for find roots start by bracketing the root
- Consider the cumulative distribution function (CDF) of a standard normal distribution

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right)\ du$$

- Goal: Find first quartile that is the value of z such that $\phi(z) = 0.25$
- Start up writing equation
- Trial and error
- Python can help

Statistical Functions in Python

- General Statistical Functions
- Normal Distribution

Python Code for Statistical Functions

```
from scipy import stats
p=0.25

x=float(input("Enter the value of x: "))
fx=stats.norm.cdf(x)

print("for x={}, f({})={}".format(x,x,fx))

print("{}-th percentile={}".format(p,stats.norm.ppf(p)))
```

Bisection Method

1. Identify an interval *[a,b]* such that either a or b overshoots the mark while the other undershoots it.
2. Calculate the midpoint, m, of the identified interval
3. If a and m both overshoot or both undershoot the mark, the desired value lies in *[m,b]*
4. If b and m both overshoots or both undershoot the mark, the desired value lies in *[a,m]*
5. Return to step 2 using the newly identified interval

Source: textbook, page 41

Simple Pseudocode

Step 1: Set $L = f(a)$;

Step 2: Set $m = \frac{a+b}{2}$; $M = f(m)$;

Step 3: If $LM < 0$ then set $b = m$; else set $a = m$ and $L = M$;

Step 4: Go to Step 2.

Source: textbook, page 43

Python code for Simple Pseudocode

```python
#from scipy import stats
def f(x):
    return stats.norm.cdf(x)-.25
# initialize code
a=-0.9
b=0.9
i=1
L=f(a)
m=(a+b)/2.0 # discuss divisor
M=f(m)
while True:
    if L*M<0.0:
        b=m
    else:
        a=m
        L=M
    m=(a+b)/2.0
```

Pseudocode 2

Assumptions: f is continuous on $[a, b]$. $f(a)$ and $f(b)$ have opposite signs.

Input: Interval $[a, b]$; function f; desired accuracy tol.

Step 1: Set $N = \left\lceil \frac{\ln(b-a)-\ln(tol)}{\ln 2} \right\rceil$; $L = f(a)$;

Step 2: For $j = 1 \ldots N$ do Steps 3-5:

Step 3: Set $m = \frac{a+b}{2}$; $M = f(m)$;

Step 4: If $M = 0$ then return m;

Step 5: If $LM < 0$ then set $b = m$; else set $a = m$ and $L = M$;

Output: Approximation m within tol of exact root or message of failure.

Source: textbook, page 43

Python Code for Pseudocode 2

```
# Pseudo-code 2
from scipy import stats
import math
def f(x):
    return stats.norm.cdf(x)-.25
# initialize code
a=-3.0
b=0.0
tol=0.0005
N=100
err=math.fabs(b-a)
L=f(a)
for i in range(0,N+1):
    m=(a+b)/2.0
    M=f(m)
    err=err/2.0
    if (M==0) or (err<tol):
```

Software

- It should be apparent that having software to follow the lectures is very helpful
  - Instructions for installing Anaconda Python
  - Instructions for creating Conda Environment