Section 1

MANE 3351

Lecture 12

Classroom Management

Agenda

- Test 1 not graded; some people still need to take
- Homework 3 Assignment
- Lab 6 Assignme
- Secant Method; last root finding method

Calendar

Date	Lecture	Lab
10/7	Secant Method, Homework 3	Lab 6
10/9	Trapezoid Rule	Lab 6, continued
10/14	Simpson's Rule, Homework 4	Lab 7
10/16	Romberg Integration	Lab 7, continued
10/21	No Class, Dr. Timmer on ABET Visit	No Class, Dr. Timmer on ABET Visit
10/23	Gaussian Quadrature, Homework 5	Lab 8
10/28	Numerical Differentiation (not on Test 2)	Lab 8, continued
11/4	Numerical Integration (not on Test 2)	Lab 9

Resources

Handouts

- Lecture 12 Slides
- Lecture 12 Marked Slides

Lecture 12 Content

- Today's topic is the secant method.
- The secant method does not utilize derivative information as Newton's method does.
- The secant method is also similar to the false position method.
- The secant method is the last root finding method covered

Limit Definition of the Derivative

Recall the limit definition of the derivative[a]

LIMIT DEFINITION OF THE DERIVATIVE

Once we know the most basic differentiation formulas and rules, we compute new derivatives using what we already know. We rarely think back to where the basic formulas and rules originated.

The geometric meaning of the derivative

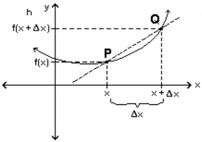
$$f'(x) = \frac{df(x)}{dx}$$

is the slope of the line tangent to $y = f(x)$ at x.

Let's look for this slope at P:

The **secant** line through P and Q has slope

$$\frac{f(x + \Delta x) - f(x)}{(x + \Delta x) - x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$



We can approximate the **tangent** line through P by moving Q towards P, decreasing Δx. In the limit as $\Delta x \to 0$, we get the tangent line through P with slope

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

We define

Geometric Inspiration

Cheney and Kincaid (2004)[a] demonstrate the geometric inspiration for secant method

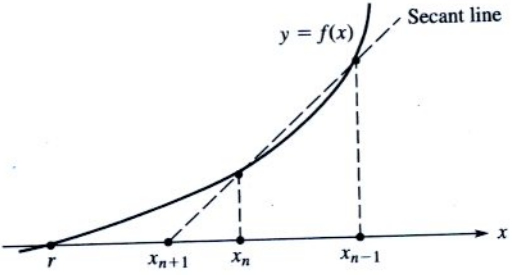


FIGURE 3.6
Secant method

Figure 2: Secant method

[a] Cheny, W., and Kincaid, D., (2004), *Numerical Mathematics and Computer, 5th edition*

Secant Method

- The formula is simply

$$x_{n+1} = x_n - g(x_n)\frac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})}$$

Example Problem Used for Newton's Method

- Consider a new function, $f(x) = e^x + 2^{-x} + 2\cos(x) - 6 = 0$

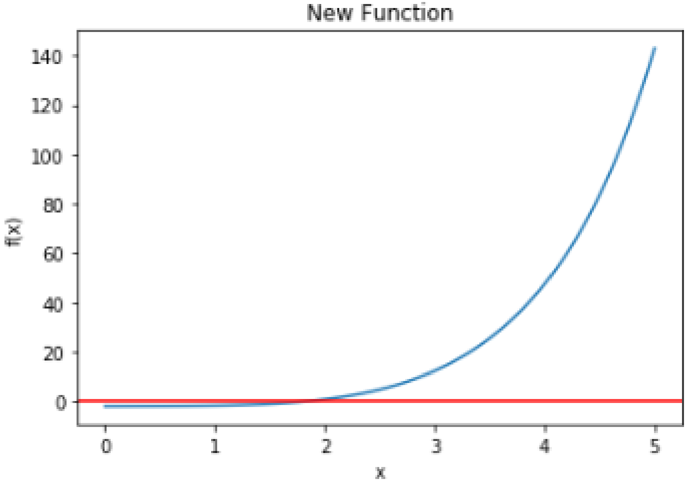


Figure 3: Example Function for Newton's method

Pseudo-code

Brin (2020)[a] provides the following pseudo-code

Secant Method (pseudo-code)

A straightforward implementation of the secant method can easily be inefficient due to the number of times g appears in formula on page 67. The pseudo-code below takes great care not to compute each value of g more than once. If it seems more complicated than necessary, this is likely the source of the complication.

Assumptions: g has a root at $\hat{x}$. g is differentiable in a neighborhood of $\hat{x}$. x_0 and x_1 are sufficiently close to $\hat{x}$.

Input: Initial values x_0 and x_1; function g; desired accuracy tol; maximum number of iterations N.

Step 1: Set $y_0 = g(x_0)$; $y_1 = g(x_1)$

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

- **Step 3:** Set $x = x_1 - y_1 \frac{x_1 - x_0}{y_1 - y_0}$;
- **Step 4:** If $|x - x_1| \leq tol$ then return x;
- **Step 5:** Set $x_0 = x_1$; $y_0 = y_1$; $x_1 = x$; $y_1 = g(x_1)$

Step 6: Print "Method failed. Maximum iterations exceeded."

Output: Approximation x near exact fixed point, or message of failure.

Figure 4: Secant Method pseud-ocode

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

Convergence

- Brin (2020)[a] study the performance of Secant Method
- The secant method converges with order $\frac{1+\sqrt{5}}{2} = 1.62$
- Not quite as fast as Newton's Method (quadratic)

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

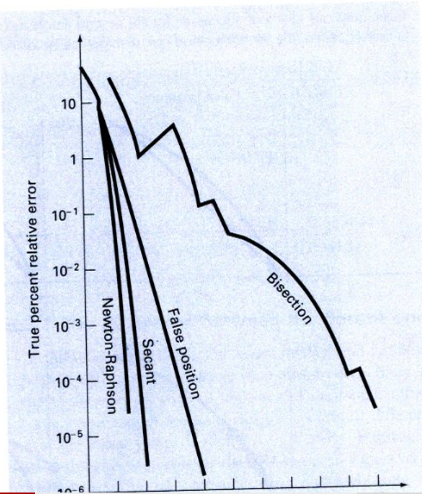
Similarity to False Position

- Secant method: $x_{n+1} = x_n - g(x_n)\frac{x_n - x_{n-1}}{g(x_n) - g(x_{n-1})}$
- False Position method: $x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$
- Secant method is not guaranteed to bracket result when starting

Speed of Convergence

Chapra and Canale (2015)[a] compared the performance of various root finding methods

FIGURE 6.9
Comparison of the true percent relative errors ε_t for the methods to determine the roots of $f(x) = e^{-x} - x$.



Secant Code

```
import math
def f(x):
    return (math.exp(x)+2**(-x)+2*math.cos(x)-6)

N=100
tol=0.0005
x0=6.0
x1=5.0
# step 1
y0=f(x0)
y1=f(x1)
# counter is additional
counter=0
for j in range(N+1):
    counter=counter+1
    # step 3
    x=x1-y1*((x1-x0)/(y1-y0))
    if math.fabs(x-x1)<tol:
```