Lecture 10 Root Finding using Open Methods
Dr. Tony Cahill

Objectives

- Open methods
  - Fixed point iteration
  - Newton-Raphson
  - Modified secant method

Open Methods vs. Bracketing Methods

- Bracketing methods
  - Root is located within the lower and upper bound.
  - Pro: Always converge to the solution
  - Con: Relatively slow
- Open methods
  - Usually starting from a single starting point (initial estimate)
  - Iteratively finding a new estimate
  - Pro: Faster convergence (when they work)
  - Con: Sometimes diverge from the true root

Single fixed point iteration

- Idea: $f(x)=0$ can be rewritten in a different form: $x=g(x)$. If $x_r$ is the root, i.e., $f(x_r)=0$, then $x_r=g(x_r)$
- Iteration process:
  - The single fixed point iterations starts from an initial guess of the root $x_0$. $x_0$ is usually not the root, i.e., $x_0 \neq g(x_0)$.
  - $g(x_0)$ can be used as the new estimate of the root: $x_1=g(x_0)$
  - If $x_1$ is quite different from $x_0$, then $g(x_1)$ is evaluated to get the new estimate of the root, and the process is repeated: $x_{i+1}=g(x_i)$.
  - The iteration stops when $x_{i+1}$ is sufficiently close to $x_i$.

Example

Find a root for $f(x)=e^{-x} - x$ using fixed point iteration.

Solution: rearrange the above function to $x=g(x)$ form.

$x = e^{-x}$

Thus, $g(x)=e^{-x}$

Check the result:

```matlab
>> f=@(x)exp(-x);
>> [root,ea]=fixpoint(f,0.5,1.0)
Solved in 11 iterations
ans =
0.5684
root =
0.5684
ea =
0.6244
```
Graphical explanation of the fixed point iteration method

From Chapra, Figure 6.3

Newton-Raphson method

- First conceived by Newton (1671) and Joseph Raphson (1690). Generalized by Thomas Simpson (1740) to modern forms.
- Very widely used
- Fast convergence (quadratic convergence)
- Need to evaluate first derivative of f(x)
- Also used in optimization

Newton-Raphson method

\[ f'(x) = \frac{f(x) - 0}{x - x_{n}} \]

Since:

\[ f'(x) = f(x) - x \]

Rearrange:

\[ x_{n+1} = x - \frac{f(x)}{f'(x)} \]

\text{FIGURE 6.4}

Example

Find a root for \( f(x) = e^{-x} - x \) using N-R method.

Solution: find \( f'(x) \) first: \( f'(x) = -e^{-x} - 1 \)

```matlab
>> f=@(x)exp(-x)-x;
>> df=@(x)exp(-x)-1;
>> [root,es]=newtonraphson(f,df,0,1.0)
Solved in 3 iterations
root =
   0.5671
es =
   0.1647
```

Problems with N-R method

\[ \text{Rate of convergence: } R_n = \left( \frac{f'(x)}{2f''(x)} \right)^n \]

Thus, "quadratic convergence"
The Modified Secant Method

- Evaluating $f'(x)$ is not always possible, thus, we need to a method to approximate $f'(x)$. One way to do this is:
  \[ f'(x) = \frac{f(x + \delta x) - f(x)}{\delta x} \]

  Thus, you get the iteration equation for the modified secant method
  \[ x_{i+1} = x_i - \frac{f(x_i)\delta x}{f(x_i + \delta x) - f(x_i)} \]

The Modified Secant Method

- Advantage:
  - No need for a separate derivative input
  - Especially useful when $f'(x)$ cannot be easily determined
- Difficulty:
  - Choose of $\delta x$ is not automatic.
    - Too small – roundoff error may dominate and lead to wrong results.
    - Too big – may become divergent

Exercise

- Program the modified secant method based on the N-R code.