3 Solving Non-linear Equations

Dr. Jeongho Ahn

Department of Mathematics & Statistics

ASU

Outline of Chapter 3

1. The Bisection Method
2. Newton's Method
3. Secant Method
4. Fixed Point Iteration
5. Ill-Behaving Rootfinding Problems

- The main idea of the bisection method is based on the Intermediate Value Theorem.

Theorem

IVT
*Suppose that f is **continuous** on $[a, b]$ and y is a number between $f(a)$ and $f(b)$. Then there exists at least one number $c \in (a, b)$ such that $f(c) = y$.*

- We can consider the following Theorem, by applying the IVT.

Theorem

*Let f be a **continuous** function on $[a, b]$ such that $f(a)f(b) < 0$. Then f has at least one root $c \in (a, b)$.*

- The previous Theorem provides the bisection algorithm:

  Given interval $[a, b]$ *and* $\varepsilon > 0$ *// ε is a stopping criterion*

  if $f(a)f(b) \geq 0$, *then stop;* $c = (a+b)/2$;

  while $|b-a| > \varepsilon$

  $c \leftarrow (a+b)/2$; // $c = a+(b-a)/2$

  compute $f(c)$

  if $f(a)f(c) < 0$

  $a \leftarrow c$; $f(a) \leftarrow f(c)$;

  else

  $b \leftarrow c$; $f(b) \leftarrow f(c)$;

  end if

  end while

  $x^* \leftarrow (a+b)/2$ //x^* *is an approximate root*

- Here is a Scilab code for the bisection method

```
// Input: x0 (a right end point of the interval, i.e., x0=b)
          tol_b(stopping criterion)
          given_fx (the function that you test)
// Output: sol(approximation of the exact solution)
           iter_count(the number of iterations)
           error_bd(the error bound)
// bis is the name of the function
// You can test any continuous functions
// The main code will be written in the next page.
```

- **function** [sol, iter_count, error_bd]=bis(x0,tol_b,given_fx)

```
a=x0-20; b=x0; fa = given_fx(a); fb = given_fx(b);
if fa*fb>0
    disp ('The bisect method does not work') //Display text
    return
end
c = (a+b)/2; iter_count=0;
while b-c>tol_b
    iter_count = iter_count+1; fc= given_fx(c);
    if fc*fb <= 0
        a = c; fa = fc;
    else
        b=c; fb=fc;
    end
        c=(a+b)/2;
end
error_bd = b-c; sol = c;
```

endfunction

- // Here is a code for the test functions
function y = test_func1(x)
y=x.*log(x+2)/3 + sin(x)/2-2;
endfunction
- How to implement your code on Scilab?

1. **Do not copy and paste my code!** You need to write the code and two functions in the Scilab text editor.
2. Load your file in the main Scilab window(Scilab Command Window).
3. Type [sol, iter_count, error_bd] = bis(22,10^-8,test_func1) to get the result. Note that sometimes it would be hard to determine the initial point x0.
4. Hit the enter key: you will see the following results.
error_bd = 9.313D-09
iter_count = 30.
sol = 3.9782445

- Convergence Analysis

Definition

Sequence x_n is said to be **linearly convergent** to a limit x if there is $C \in [0,1)$ such that

$$|x_{n+1} - x| \leq C\,|x_n - x| \quad \text{for } n \geq 1.$$

Alternatively, sequence x_n is said to be **linearly convergent** to a limit x with rate S if

$$\lim_{n\to\infty} \frac{|x_{n+1} - x|}{|x_n - x|} = S < 1$$

1. c_n (midpoints by the b. m.) converges linearly to a solution x.
2. Suppose that $|x - c_n| \leq \varepsilon$. How many iterations will be necessary to be satisfied with the assumption?
3. **Advantage**: the bisection method is guaranteed to converge (linearly) to a solution if $f \in C[a,b]$.
4. **Dis...**: it converges very slowly, compared to N. M. and S. M.