MT308 Numerical Analysis

Girne American University
Faculty of Engineering
Spring 2026
Dr. Kamil Dimililer

1

MATLAB Functions

- Built-in Functions: Pre-defined functions
- User-Defined Functions
  - **Function Files:** Saved in a separate .m file

```
%Within squarenum.m file
function y = squarenum(x)
 y = x^2;
end
```

  - **Anonymous Functions Function** Syntax

```
f = @(x) x^2;
```

  - **Nested Functions** are defined inside another function, which can access variables from the parent function
  - **Local Functions** are defined at the end of a script file and are accessible only within that file

2

Anonymous Functions in MATLAB

An anonymous function is a function that is not stored in a program file but is associated with a variable of type **function_handle**. Anonymous functions can accept multiple inputs and return one output. They can contain only a single executable statement.

Function Syntax : **f = @(x) expression**

- `@` creates a function handle
- `x` = input argument
- `expression` = output expression

Examples:

```
>> f = @(x) x^2 + 3*x + 1;
>> f(2)
  ans = 11

>> go = @(x,y) x^2 + y^2;
>> go(2,3)
  ans = 13

>> cal = @(x) sin(x) + exp(x);
>> cal(2*pi/4)
  ans = 5.8105
```

3

A **function handle** is a MATLAB data type that **represents a function.** A typical use of function handles is to pass a function to another function. For example, you can use function handles as input arguments to functions that evaluate mathematical expressions over a range of values.

The symbol @ can be used to obtain a function handle for a built-in function or a function defined by the function keyword.

The names of the anonymous functions are their handles

Function	Function Handle
sin(x)	@sin
function y = cubicPoly(x) y = x.^3 + x.^2 + x + 1; end	@cubicPoly
cubPol=@(x) x.^3 + x.^2 + x + 1;	cubPol

4

Solving nonlinear equations

Let f(x) to be a nonlinear function of the variable x. The equation **f(x)=0** can be solved by different methods.

Examples:

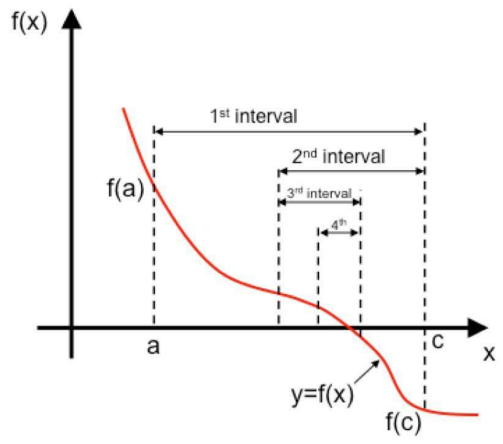
$$x^2 - 3x + 4 = 0$$
$$x + 5 = 0$$
$$\cos(x) - \tan(x) + 0.5 = 0$$
$$\ln(x - 1) + \exp(3x) + 2 = 0$$

In some cases, we may find analytical solutions. But in many practical cases either the analytical solutions are not available, or it is very difficult to find them. Therefore, **numerical methods** are required to solve such problems.

5

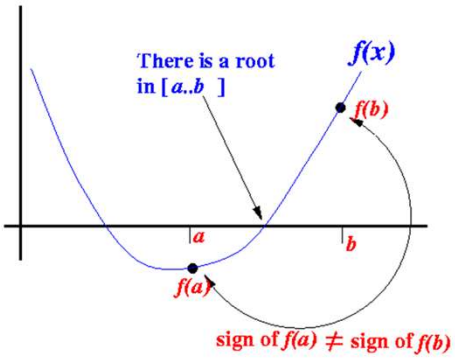
Bisection Method

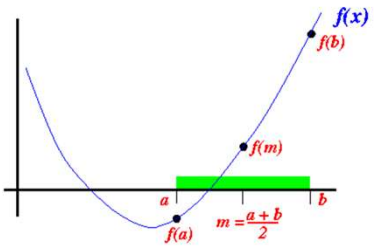
- The **Bisection Method** is a numerical **root-finding technique** used to solve equations. It works by **dividing an interval into smaller intervals** and narrowing down the range of possible solutions **iteratively**.
- The process of repeatedly dividing a set in half to find a solution to a problem is known as **binary search**.



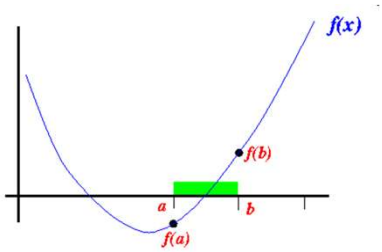
6

The Bisection Method



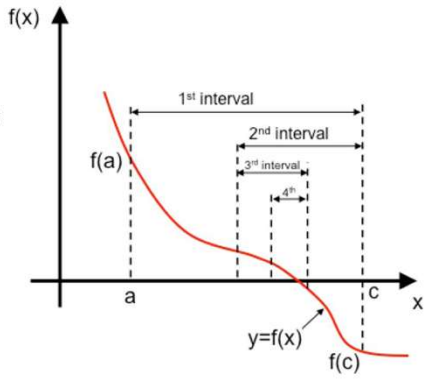


If a function f(x) is continuous on the interval [a..b] and sign of f(a) ≠ sign of f(b), then there is a value c ∈ [a..b] such that: f(c) = 0 i.e., there is a root c in the interval [a..b]



7

Steps of the Bisection Method

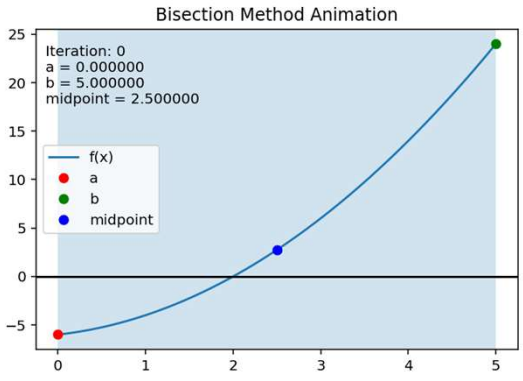


1. **Bracketing the Root:**
   Start with an interval [a, b] where the function f(x) changes sign, meaning f(a) and f(b) have opposite signs. This guarantees at least one root exists within that interval, according to the intermediate value theorem.
2. **Finding the Midpoint:**
   Calculate the midpoint of the interval: x = (a + b) / 2.
3. **Refining the Interval:**
   If f(x) = 0, then x is the root.
   If f(x) and f(a) have the same sign, then the root lies in the interval [x, b]. Replace a with x.
   If f(x) and f(b) have the same sign, then the root lies in the interval [a, x]. Replace b with x.
4. **Repeating the Process**:
   Repeat steps 2 and 3 with the new interval until the desired level of accuracy is reached.

8

Example: Use the bisection method to find the root of f(x)=$x^2 + x - 6 = 0$

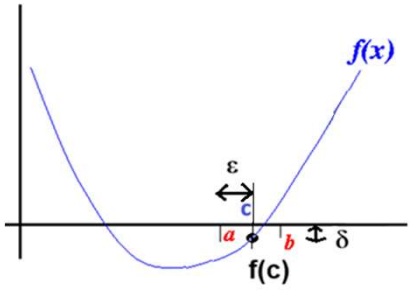
Note: f(x)=(x-2)(x+3) and it has two roots at x=2 and x=-3.



a	x	b	f(a)	f(x)	f(b)	b-a
0	2.5	5	-6	2.75	24	5
0	1.25	2.5	-6	-3.1875	2.75	2.5
1.25	1.875	2.5	-3.1875	-0.60938	2.75	1.25
1.875	2.1875	2.5	-0.60938	0.972656	2.75	0.625
1.875	2.03125	2.1875	-0.60938	0.157227	0.972656	0.3125
1.875	1.953125	2.03125	-0.60938	-0.23218	0.157227	0.15625
1.953125	1.992188	2.03125	-0.23218	-0.039	0.157227	0.078125
1.992188	2.011719	2.03125	-0.039	0.058731	0.157227	0.039063
1.992188	2.001953	2.011719	-0.039	0.009769	0.058731	0.019531
1.992188	1.99707	2.001953	-0.039	-0.01464	0.009769	0.009766
1.99707	1.999512	2.001953	-0.01464	-0.00244	0.009769	0.004883
1.999512	2.000732	2.001953	-0.00244	0.003663	0.009769	0.002441
1.999512	2.000122	2.000732	-0.00244	0.00061	0.003665	0.001221
1.999512	1.999817	2.000122	-0.00244	-0.00092	0.00061	0.00061
1.999817	1.999969	2.000122	-0.00092	-0.00015	0.00061	0.000305
1.999969	2.000046	2.000122	-0.00015	0.000229	0.00061	0.000153
1.999969	2.000008	2.000046	-0.00015	3.81E-05	0.000229	7.63E-05
1.999969	1.999989	2.000008	-0.00015	-5.7E-05	3.81E-05	3.81E-05
1.999989	1.999998	2.000008	-5.7E-05	-9.5E-06	3.81E-05	1.91E-05
1.999998	2.000003	2.000008	-9.5E-06	1.43E-05	3.81E-05	9.54E-06

9

Note 1: There are two different options to define the error when we use the bracketing methods: (1) The function value (δ=|f(c)|); and (2) half of the Interval width (ε)



Note 2: The error made is less than half of the bracket at each iteration. Since the bracket is divided to two, the error at the Nth iteration is $\boldsymbol{\varepsilon = \frac{b-a}{2^N}}$

If we get the logarithm of each side $\ln(\varepsilon) = ln\left(\frac{b-a}{2^N}\right)$, the following expressions can be used to calculate the number of iterations required to achieve the desired error level:

$$N \cong \left(\frac{\ln(b-a)-\ln(\varepsilon)}{\ln(2)}\right) \text{ or } N \cong log_2\left(\frac{b-a}{\varepsilon}\right)$$

10

```
function c = mybisection(f, a, b, tol)
% MT308, A function for finding the root of a function using
%the bisection method. Use mybisection(f, a, b, tol)
%where f is the name of the function, a and b are the lower and
% upper limits of the bracket, and tol is the value of f(x)
% to be considered as zero.
    if f(a) * f(b) >= 0
        error('Invalid interval')
    end
    c = (a + b) / 2;
    while abs(f(c)) >= tol
        c = (a + b) / 2;
        if f(c) == 0
            break;
        elseif f(a) * f(c) < 0
            b = c;
        else
            a = c;
        end
    end
end
```

Calling the mybisection function from the command window

```
>> f=@(x) x^2+x-6;
>> a=mybisection(f,0,5,1E-3)
a = 2.0001
>> a=mybisection(f,0,5,1E-7)
a = 2.0000
>> a=mybisection(f,-8,1,1E-7)
a = -3.0000
>> g=@(x) cos(x)-tan(x)-0.5;
>> a=mybisection(g,-pi/4,pi/4,1E-7)
a = 0.3989
>>
```

11

Key features of the Bisection Method

- **Reliable**: It is guaranteed to converge to a root if a sign change exists in the initial interval.

  But it is required to find an appropriate **initial interval** where the function changes sign. It struggles with multiple roots within the interval, only finding one at a time
- **Slow Convergence**: The method converges linearly, which means it may take more iterations to reach the desired accuracy than other methods.
- **Simple Implementation**: It's relatively easy to understand and implement.

12

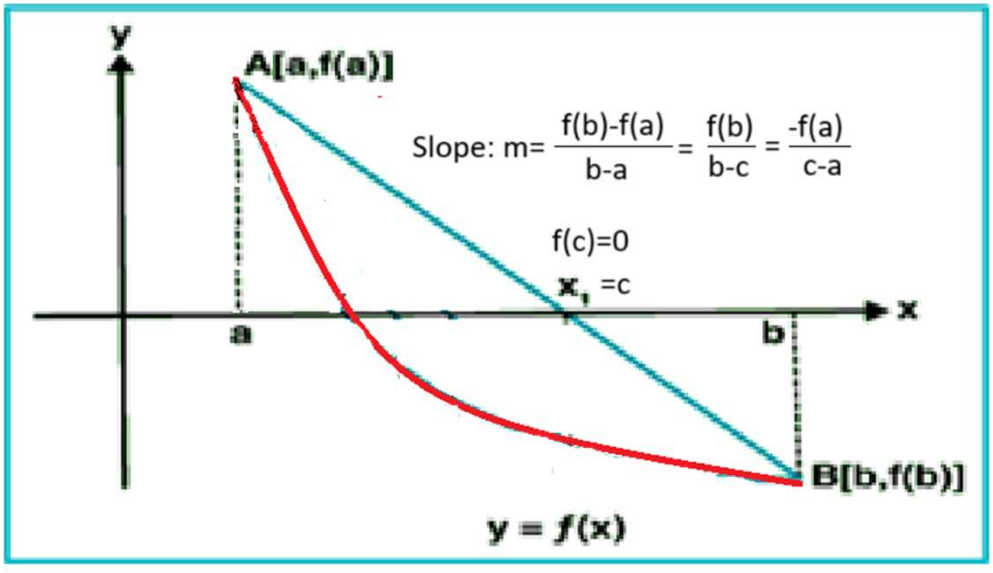
False-Position (Regula Falsi) Method

- The False-Position (or Regula-Falsi) Method is similar to the bisection method, with a slightly improved convergence rate.
- Upper and lower bounds of the root are used to locate a usually better estimate.
- If robustness and reliability are more important than speed, the Bisection Method is preferred.
- Regula Falsi method is better if you want potentially faster convergence. But it is not guaranteed for all problems.
- The convergence rate of the Regula-Falsi method depends strongly on the function's behaviour on the interval: When the function behaves nearly linearly near the root, the Regula-Falsi method converges faster than the Bisection method.

13

13

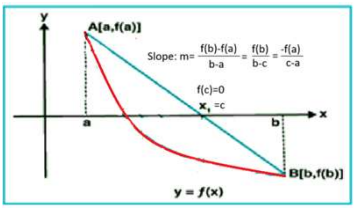
False-Position (Regula Falsi) Method



14

14

False-Position Method (From the figure)



$$\frac{1}{m}=\frac{b-a}{f(b)-f(a)}=\frac{b-c}{f(b)}$$

$$c=b-\frac{f(b)(b-a)}{f(b)-f(a)}$$

or

$$c=\frac{f(b)a-f(a)b}{f(b)-f(a)}$$

$$\frac{1}{m}=\frac{b-a}{f(b)-f(a)}=\frac{a-c}{f(a)}$$

$$c=\mathrm{a}-\frac{f(a)(b-a)}{f(b)-f(a)}$$

or

$$c=\frac{f(b)a-f(a)b}{f(b)-f(a)}$$

The Same !!

15

15

```
function c = myregula(f, a, b, tol)
% MT308, A function for finding the root of a function using
%the false position method. Use myregula(f, a, b, tol)
%where f is the name of the function, a and b are the lower and
% upper limits of the bracket, and tol is the value of f(x)
% to be considered as zero.
    if f(a) * f(b) >= 0
        error('Invalid interval')
    end
    c = (f(b)*a-f(a)*b)/(f(b)-f(a));
    while abs(f(c)) >= tol
        c = (f(b)*a-f(a)*b)/(f(b)-f(a));
        if f(c) == 0
            break;
        elseif f(a) * f(c) < 0
            b = c;
        else
            a = c;
        end
    end
end
```

Command window exercises

```
>> f=@(x) x^2+x-6;
>> myregula(f,0,6,1E-5)
ans = 2.0000
>> h = @(x) 3*sin(x) -0.02* exp(x);
>> myregula(h,0.5,6,1E-5)
ans = 3.0064
```

16

	Bisection Method	Regula Falsi Method
Approach	Uses the midpoint: $c = \frac{a+b}{2}$	Uses a linear interpolation: $c = \frac{f(b)a - f(a)b}{f(b) - f(a)}$
Convergence	Guaranteed, but **slower** (linear convergence)	Often **faster** than bisection, but ***not guaranteed to improve speed in all cases***
New position and interval	Only uses the sign of the function value	Uses both sign and **magnitude** of function values
Root Location Guarantee	If the initial interval is well-defined and the function is continuous, it ***always converges to a root***	Also ***guarantees convergence under the same*** conditions, but ***can stall on one side***
Speed	Slower but reliable	***Faster in some cases***, but can suffer from **slow convergence** if the function is not well-behaved
Implementation Complexity	Very simple	Slightly more complex due to interpolation formula

17

Example: Find a root of $f(x) = x^2 + 2\sqrt{3}x - 9$ within the interval $0 \leq x \leq 5$ by using (a) The bisection Method; (b) The Regula-Falsi Method. (Show only the first 3 iterations)

In written exams, you will need to solve this type of question using your calculator. Therefore, it is important to learn how to use your calculators, and you must bring your own calculators to all written exams.

BISECTION METHOD

it	a	c	b	f(a)	f(c)	f(b)	Error
0	0	2.5	5	-9	5.91	33.32	5.91
1	0	1.25	2.5	-9	-3.11	5.91	3.11
2	1.25	1.875	2.5	-3.11	1.01	5.91	1.01
3	1.25	1.5625	1.875	-3.11	-1.15	1.01	1.15

FALSE POSITION METHOD

it	a	c	b	f(a)	f(c)	f(b)	Error
0	0	1.063327	5	-9	-4.19	33.32	4.19
1	1.063327	1.503067	5	-4.19	-1.53	33.32	1.53
2	1.503067	1.656591	5	-1.53	-0.52	33.32	0.52
3	1.656591	1.707967	5	-0.52	-0.17	33.32	0.17

18

Example: Use the False Position method to find the the root of f(x)=log(x+ 3)) with an initial interval of [-2.5, 20]. Use a tolerance value of 1.0E-6

Iter	a	b	c	f(a)	f(c)	b-a
1	-2.5	20	**1.573458**	-0.69315	1.52E+00	22.5
2	-2.5	**1.573458**	-1.22437	-0.69315	5.74E-01	4.073458
3	-2.5	-1.22437	-1.8023	-0.69315	1.80E-01	1.275632
4	-2.5	-1.8023	-1.94639	-0.69315	5.22E-02	0.697702
5	-2.5	-1.94639	-1.98518	-0.69315	1.47E-02	0.553614
6	-2.5	-1.98518	-1.99588	-0.69315	4.11E-03	0.514824
7	-2.5	-1.99588	**-1.99885**	-0.69315	1.15E-03	0.504122
8	-2.5	**-1.99885**	-1.99968	-0.69315	3.20E-04	0.501148
9	-2.5	-1.99968	-1.99991	-0.69315	8.91	
10	-2.5	-1.99991	-1.99998	-0.69315	2.48	
11	-2.5			-0.69315	6.92	
12	-2.5			-0.69315	1.93	
13	-2.5			-0.69315	5.37	

Since F(b) and f(c) have opposite signs in all iterations, the lower bound a remained unchanged throughout the process

Stopped! since the f(c) value is less then the tolerance

5.37E-7<1.0E-6

19

Compare Regula-Falsi and Bisection method to solve the equation log(x+ 3)=0 (i.e. To find the root of f(x)=log(x+ 3)) with an initial interval of [-2.5, 20]. Use a tolerance value of **1.0E-6**.

	Bisection		Regula Falsi	
Iter	c	f(c)	c	f(c)
1	8.75	2.46E+00	1.573458	1.52E+00
2	3.125	1.81E+00	-1.224368	5.74E-01
3	0.3125	1.20E+00	-1.802298	1.80E-01
4	-1.09375	6.45E-01	-1.946386	5.22E-02
5	-1.796875	1.85E-01	-1.985176	1.47E-02
6	-2.148438	-1.61E-01	-1.995878	4.11E-03
7	-1.972656	2.70E-02	-1.998852	1.15E-03
8	-2.060547	-6.2		E-04
9	-2.016602	-1.6		E-05
10	-1.994629	5.3		E-05
11	-2.005615	-5.6		E-06
12	-2.000122	-1.2		E-06
13	-1.997375	2.62E-03	-1.999999	**1.00E-06**

For this example, the Bisection method needs to continue until the 24th iteration to reach the same **f(c)** value that is obtained at the 13th iteration by the False position.

20

Some useful MATLAB command for converting real numbers to integer	
Y = ceil (X)	Return the smallest integer not less than X
Y = floor (X)	Return the largest integer not greater than X
Y = round (X)	Return the integer nearest to X
Y = fix (X)	Truncate the fractional portion of X and return the integer portion

Assignment 2 (Section 1)

- Modify the MATLAB function the "mybisection" so that (1) its name is your first name, (2) Arguments are f, a, b, and eps ($\boldsymbol{\varepsilon}$); (3) Use a for loop instead of the while loop using the following expression,

$$N \cong \left(\frac{\ln(b-a)-\mathrm{l}\ (\varepsilon)}{\ln(2)}\right)$$

- Write a command window expression to find the root of

$$f(x)=2\ln(x-1)-4\exp(x/10)+5$$

Within the interval 1.5<x<4 where $\varepsilon = 10^{-4}$

- Upload the m file and the image of the command window that you solve the problem

21

Some useful MATLAB command for converting real numbers to integer	
Y = ceil (X)	Return the smallest integer not less than X
Y = floor (X)	Return the largest integer not greater than X
Y = round (X)	Return the integer nearest to X
Y = fix (X)	Truncate the fractional portion of X and return the integer portion

Assignment 2 (Section 2)

- Modify the MATLAB function the "mybisection" so that (1) its name is your surname, (2) Arguments are f, a, b, and eps ($\boldsymbol{\varepsilon}$); (3) Use a for loop instead of the while loop using the following expression,

$$N \cong \left(\frac{\ln(b-a)-\ln(\varepsilon)}{\ln(2)}\right)$$

- Write a command window expression to find the root of

$$f(x)=\ln(2x+7)-\exp(x/2)$$

Within the interval -3<x<-2 where $\varepsilon = 10^{-6}$

- Upload the m file and the image of the command window that you solve the problem

22