

- Numerical method

- Summer 2020

- Dr. Naim al-fakeer

- ~~XXXXXXXXXX~~



- Measuring Error

• Why!

- ① to determine the accuracy of numerical result
- ② to develop stopping Criteria for iterative algorithms

A: True Error

$$\text{True Error} = \text{True value} - \text{Approximate value}$$

B: Relative True Error

$$\text{Relative Error} = \frac{\text{True Error}}{\text{True value}}$$

C: Approximate Error

$$\text{Approximate Error} = \text{present approximation} - \text{previous approximation}$$

D: Relative Approximate Error

$$\text{Relative Approximate Error} = \frac{\text{Approximate Error}}{\text{Present approximation}}$$



Part 1

⊗ Numerical methods

- algorithms that are used to obtain numerical solution of mathematical problem.

• Why!

- no analytical solution exists
- analytical solution is difficult to obtain.

⊗ methods for solving nonlinear equations:-

- Bisection method
- Newton - Raphson method
- secant method



1. Bisection method:-

Theorem:-

An equation $f(x)=0$, where $f(x)$ is a real, continuous function has at least one root between x_1 and x_2 if $f(x_1) \cdot f(x_2) < 0$

Notes:

- ① at least one root exist between the ^{two} ~~root~~ points.
if the function is real, continuous and change sign
- ② if $f(x)$ doesn't change sign between two points, roots of the equation may still exist between the two point
- ③ if the function doesn't change sign between points there may not be any roots for the equation
- ④ if the sign between two point change, more than one root for equation may exist.

④ How to solve by Bisection method:-

- ① specific x_l and x_u
- ② evaluate $f(x_l)$ and $f(x_u)$
- ③ check sign $f(x_l) \neq f(x_u)$
- ④ evaluate $x_3 \rightarrow x_3 = \frac{x_l + x_u}{2}$
- ⑤ if $f(x_3) * f(x_l) > 0$ $x_l = x_3$
if $f(x_3) * f(x_u) < 0$ $x_u = x_3$
- ⑥ Check the error $\leq \text{tol}$ stop else repeat from ④

④ Coding by matlab:-

```
y = inline('f(x)');
```

```
x1 = a;
```

```
x2 = b;
```

```
x3 = (x1 + x2) / 2;
```

```
tol = c;
```

```
while abs((x2 - x1) / x3) > tol
```

```
if y(x1) * y(x2) > 0
```

```
    x1 = x3
```

```
    x2 = x2
```

```
else
```

```
    x2 = x3
```

```
end
```

```
    x3 = (x1 + x2) / 2
```

```
end
```



3. Secant method

Steps of secant method

1. specific x_1 and x_2
2. evaluate $f(x_1)$ and $f(x_2)$
3. evaluate $x_3 \Rightarrow x_3 = x_1 - \frac{f(x_1) * (x_1 - x_2)}{f(x_1) - f(x_2)}$
4. evaluate $f(x_3)$
5. evaluate absolute error
6. abs error $< tol$ x_3 is the root
else $x_1 = x_2$
 $x_2 = x_3$

Coding by matlab:-

```
y = inline('f(x)');
```

```
x1 = a;
```

```
x2 = b;
```

```
x3 = x1 - (f(x1) * (x2 - x1)) / (f(x2) - f(x1));
```

```
while abs((x2 - x1) / x3) > tol
```

```
    x1 = x2;
```

```
    x2 = x3;
```

```
    x3 = x1 - (f(x1) * (x2 - x1)) / (f(x2) - f(x1));
```

```
end
```



3. Newton-Raphson method

- steps:-

- Specific x_1
- evaluate $f(x_1)$
- evaluate $f'(x_1)$
- $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$
- evaluate abs error
- abs error $> tol$ x_2 is the root
else $x_1 = x_2$ and repeat.

- Coding by matlab:-

$y = \text{inline}('f(x)');$; in new script:-

$x_1 = a;$

$\text{syms } y(x)$

$x_2 = x_1 - \left(\frac{y(x_1)}{D(x_1)} \right);$

$y(x) = f(x);$

$D = \text{diff}(y(x))$

while abs $\left(\frac{x_2 - x_1}{x_2} \right) > tol$

$x_2 = x_1 - \left(\frac{y(x_1)}{D(x_1)} \right)$

$x_1 = x_2$

end



4. False - position method

- Steps:-

- Specific x_1, x_2

- evaluate $f(x_1), f(x_2)$

- evaluate $x_3 = x_1 - \frac{f(x_1)(x_1 - x_2)}{f(x_1) - f(x_2)}$

- evaluate $f(x_3)$

- if $f(x_3) * f(x_1) > 0$ $x_1 = x_3$

else $f(x_3) * f(x_2) < 0$ $x_2 = x_3$

- Coding by matlab:-

```
y = inline(' f(x) ');
```

```
x1 = a;
```

```
x2 = b;
```

```
x3 = x1 - (y(x1)(x1 - x2)) / (y(x1) - y(x2))
```

```
while abs(x2 - x1) > tol
```

```
if y(x1) * y(x2) > 0
```

```
    x1 = x3;
```

```
else
```

```
    x2 = x3;
```

```
end
```



5. Fixed point iteration

- steps:-

- $f(x) = 0$
- $g(x) = x_1$
- evaluate $g(x_1)$
- $x_2 = g(x_1)$
- evaluate abs error
- if $\text{abs} < \text{tol}$ $x_2 = \text{root}$ else $x_1 = x_2$

- coding by matlab:-

```
y = inline('f(x)');
```

```
x1 = a;
```

```
x2 = g(x1);
```

```
while abs( $\frac{x_2 - x_1}{x_2}$ ) > tol
```

```
    x1 = x2
```

```
    x2 = g(x1)
```

```
end
```



Find root by matlab build-in function.

① Graph :-

$x = [\quad : \quad]$ interval

$y = f(x)$

$\text{plot}(x, y)$

② Root

$y = [a \quad b \quad c]$

$\leq [\text{tolerance}]$

$r = \text{root}(y)$

③ f zero

$F = \text{inline}('f(x)')$

$y = fzero(y, F)$

F : initial guess from graph.

④ Syms

$\text{syms}(y, x)$

$y = f(x)$

$P = \text{solve}(y)$



part 2.

methods to solve system of linear eq

1. Gaussian Elimination

2. Gaussian seidel

3. LU method

- steps of Gaussian elimination:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow$$

- بتفرغ المعادلات بمقدار A

- بتفرغ نتائج المعادلات بمقدار B

- بتفرغ متغيرات بمقدار X

$$B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$[A][X] = [B]$$



* Steps

① make upper matrix.

$$\begin{array}{l} R_1 \\ R_2 \\ R_3 \end{array} \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0^{①} & a_{22} & a_{23} & b_2 \\ 0^{②} & 0^{③} & a_{33} & b_3 \end{array} \right]$$

* $R_2 \rightarrow ① \quad -\left(\frac{a_{12}}{a_{11}}\right) * a_{11} + a_{12} = 0$

التحويل على السطر الثاني

وتصبح a_{12}

$\rightarrow -\left(\frac{a_{12}}{a_{11}}\right) * R_1 + R_2$

* $R_3 \rightarrow ② \quad -\left(\frac{a_{13}}{a_{11}}\right) * a_{11} + a_{13} = 0$

تصبح a_{13}

التحويل على السطر الثالث

$\rightarrow -\left(\frac{a_{13}}{a_{11}}\right) * R_1 + R_3$

* $R_3 \rightarrow ③ \quad -\left(\frac{a_{23}}{a_{22}}\right) * a_{22} + a_{23} = 0$

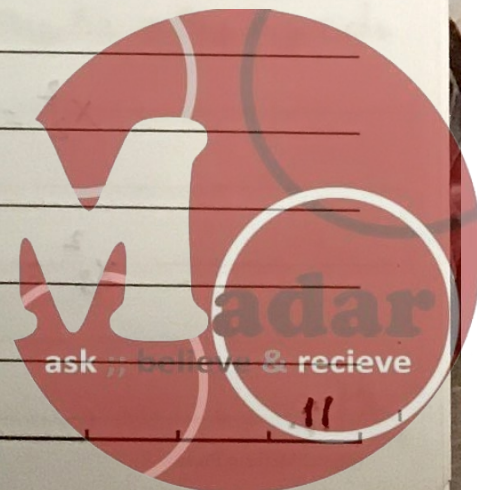
تصبح a_{23}

$\rightarrow -\left(\frac{a_{23}}{a_{22}}\right) * R_2 + R_3$

② $x_3 a_{33} = b_3$

$x_2 a_{22} + x_3 a_{23} = b_2$

$x_1 a_{11} + x_2 a_{12} + x_3 a_{13} = b_1$



Gaussian Seidl method

* steps:-

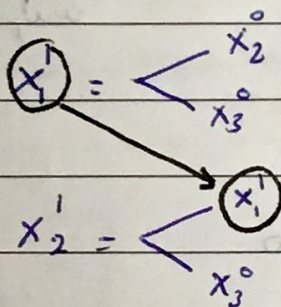
① $X_0 = [x_1^0, x_2^0, x_3^0]$ initial

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$$

$$x_2 = \frac{b_2 + a_{21}x_1 - a_{23}x_3}{a_{22}}$$

$$x_3 = \frac{b_3 - a_{31}x_1 - a_{32}x_2}{a_{33}}$$

$itr^0 = [x_1^0, x_2^0, x_3^0]$: طريقة التقريب



$$x_3^1 = \frac{b_3 - a_{31}x_1^1 - a_{32}x_2^1}{a_{33}}$$

$itr^1 = [x_1^1, x_2^1, x_3^1]$

$$x_1^2 = \frac{b_1 - a_{12}x_2^1 - a_{13}x_3^1}{a_{11}}$$

$$x_2^2 = \frac{b_2 + a_{21}x_1^2 - a_{23}x_3^1}{a_{22}}$$

$$x_3^2 = \frac{b_3 - a_{31}x_1^2 - a_{32}x_2^2}{a_{33}}$$



* Solve matrix by matlab build-in function.

1- $a = [\quad]$

$b = [\quad ; \quad ; \quad]$

$x = a \setminus b$

2- $a = [\quad]$

$b = [\quad ; \quad ; \quad]$

$c = \text{inv}(a)$

$x = c * b$

3- $\text{syms } x_1 \quad x_2$

$a = f_1$

$b = f_2$

$y = \text{solve}(f_1, f_2 \dots)$

$y = x_1$

$y = x_2$



Least squares Regression

① Curve Fitting :-

* ① Linearization

$$y = ax + b$$

$$a = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

$$b = \bar{y} - a\bar{x}$$

$$② r^2 = \frac{\sum (y_i - \bar{y})^2 - \sum (y_{calc} - y_i)^2}{\sum (y_i - \bar{y})^2}$$

- Curve fitting by matlab:-

$$x = [\quad] \quad \text{صفه}$$

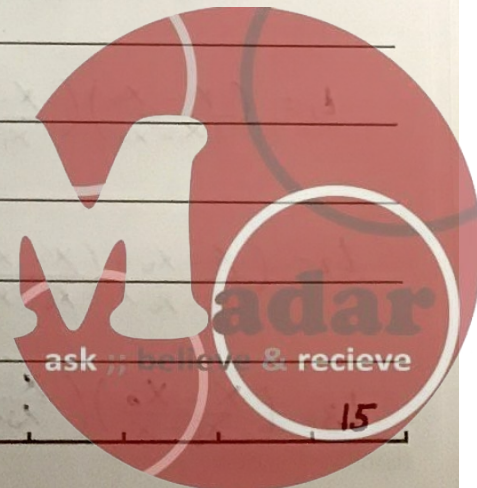
$$y = [\quad]$$

after Linearization $\begin{cases} x' = \\ y' = \end{cases}$

$$I = \text{polyfit}(x', y', n) \rightarrow \text{النتيجة}$$

$$a = I(1)$$

$$b = I(2)$$



* interpolation

1. Lagrange method

$$f(x) = L(x) f(x_n)$$

① Linear

$$f(x) = L_0(x) f(x_0) + L_1(x) f(x_1)$$

$$L_0 = \frac{x - x_1}{x_0 - x_1}$$

$$L_1 = \frac{x - x_0}{x_1 - x_0}$$

② quadratic

$$f(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2)$$

$$L_0 = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right)$$

$$L_1 = \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right)$$

$$L_2 = \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right)$$

③ Cubic

$$f(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2) + L_3(x) f(x_3)$$

$$L_0 = \left(\frac{x - x_1}{x_0 - x_1} \right) \left(\frac{x - x_2}{x_0 - x_2} \right) \left(\frac{x - x_3}{x_0 - x_3} \right)$$

$$L_1 = \left(\frac{x - x_0}{x_1 - x_0} \right) \left(\frac{x - x_2}{x_1 - x_2} \right) \left(\frac{x - x_3}{x_1 - x_3} \right)$$

$$L_2 = \left(\frac{x - x_0}{x_2 - x_0} \right) \left(\frac{x - x_1}{x_2 - x_1} \right) \left(\frac{x - x_3}{x_2 - x_3} \right)$$

$$L_3 = \left(\frac{x - x_0}{x_3 - x_0} \right) \left(\frac{x - x_1}{x_3 - x_1} \right) \left(\frac{x - x_2}{x_3 - x_2} \right)$$

Newton method:-

Linear

$$f(x) = b_0 + b_1(x - x_0)$$

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

- quadratic

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$b_2 = \frac{\frac{f(x_2) - f(x_0)}{x_2 - x_0} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

- interpolation by matlab:-

$$x = [\quad]$$

$$y = [\quad]$$

$$h = \text{interp}(x, y, g)$$

المسألة المطلوب حلها: g

① Differentiation:-

Steps:-

A - Forward: $f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$

Backward: $f'(x_i) = \frac{f(x_i) - f(x_{i-1}))}{h}$

Center: $f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$

② integration:-

1. Trapezoidal Rule

2. Simpson 1/3 Rule

1. Trapezoidal Rule:-

$$\int_a^b f(x) dx = b-a \left[\frac{f(a) + f(b)}{2} \right]$$

- multiple segment trapezoidal Rule

$$\int_a^b f(x) dx = \frac{h}{2} [f(a) + 2(f(a+h)) + f(b)]$$

$$h = \frac{b-a}{n}$$



2. Simpson 1/3 Rule

$$\int_a^b f(x) dx = \frac{b-a}{3} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \quad h = \frac{b-a}{2}$$

- multiple Simpson Rule:-

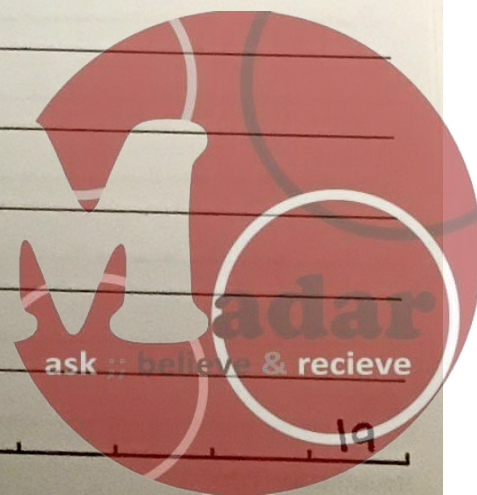
$$\int_a^b f(x) dx = \frac{b-a}{3n} \left[f(x_0) + 4 \sum_{i=\text{odd}} f(x_i) + 2 \sum_{i=\text{even}} f(x_i) + f(x_b) \right]$$

- integration by matlab

$x = [\quad]$

$y = [\quad]$

$z = \text{trapz}(x, y)$



Solving ODE

- Euler method
- Runge - Kutta 2nd method
- Runge - Kutta 4th method

- Euler method =

$$y_2 = y_1 + f'(x_1, y_1) \cdot h$$

$$h = x_2 - x_1$$

- Runge Kutta 2nd

$$y_2 = y_1 + \frac{h}{2} (k_1 + k_2)$$

$$k_1 = f(x_1, y_1)$$

$$k_2 = f(x_1 + h, y_1 + k_1 h)$$

- Runge Kutta 4th

$$y_2 = y_1 + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(x_1, y_1)$$

$$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1 h}{2}\right)$$

$$k_3 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2 h}{2}\right)$$

$$k_4 = f(x_1 + h, y_1 + k_3 h)$$

