ملخصات مدار

د. نعيم الفقير

الطرق العددية في الهندسة الكيميائية

رنيم شموط

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

True value ⇓ Calculations; Approximate value ⇓ Numerical method

* $$\boxed{\text{Relative True Error } (\epsilon_t) = \frac{\text{True Error}}{\text{True value}}}$$

* $\boxed{\text{Approximate Error } (E_a) = \text{Present Approx.} - \text{Previous Approx.}}$ ⇒ when true value not known or very difficult to obtain

* $$\boxed{\text{Relative Approx. Error } (\epsilon_a) = \frac{\text{Approximate Error}}{\text{Present Approx.}}}$$

* $\boxed{|\epsilon_a| \leq \epsilon_s}$ ⇒ no further iterations are necessary ⇒ process stoped

ϵ_s ⇓ pre-specified tolerance

* at least m significant digit are required to be correct in final answer ⇒ $\boxed{|\epsilon_a| \leq 0.5 \times 10^{+2-m} \%}$

* Accuracy ⇒ How close measured value to true value

* precision (reporoducibility) ⇒ " " " " " previous values

* inaccuracy (bias) ⇒ systematic deviation from actual value

* Imprecision (uncertainity) → Magnitude of scatter

* significant figures ⇒ indicates precision

* Bracketing methods
  - graphical
  - bisection
  - false position

* open methods
  - fixed point iteration
  - Newton
  - secant

* Two source of numerical error ⇒

Round off error ⇒ caused by represent number Approximately

Truncation error ⇒ " " truncating or approximat mathmatical Procedure.

* Methods of solving non linear equation ⇒

① Bisection
⇓
without derivative.

advantge :
① convergent
② halved with each iteration (garanteed)

Drawbacks :
① slow convergence
② initial guess close to root (slowerconvergence)

$$*\ L_K = \frac{L_0}{2^K}$$

$$\varepsilon_a \leq \frac{L_K}{x} \times 100\%$$

$$\varepsilon_a \leq \varepsilon_s$$

② Newton - Raphson

advantge :
① converges fast (quadratic)
② requires 1 guess

Drawbacks :
① Divergence at inflection point.
② Division by zero.
③ oscillations near local max. and minum.
④ Root jumping

③ secant.

advantage :
① converges fast
② require 2 guess not need to bracket root

Drawback :
① Divison by zero.
② Root jumping.

* open Methods ⇒ require single starting value of x or two starting values that do not necessarily bracket the root.

- fixed point iteration Method :

$f(x) = (\quad) = 0$ $\qquad$ $\varepsilon_a \downarrow \Rightarrow$ convergence $\Rightarrow |g'(x)| < 1$

$x = g(x)$ $\qquad$ $\varepsilon_a \uparrow \Rightarrow$ Divergence.

x	$g(x)$ (x_{new})	$\varepsilon_a\%$

Initial guess, shape ⇒ may sometime diverge.

- Newton-Raphson Method : (Most widely used), based on Taylor series expansion

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

(convenient when derivative can be evaluated Analytically)

- Secant Method : backward finite divided difference (when f' difficult to find)

$$x_{i+1} = x_i - f(x_i)\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}$$

require 2 guess, not bracketing Method

* Multiple Roots ⇒ correspond to point where function tangent to x axis.

① $u(x_i) = \dfrac{f(x_i)}{f'(x_i)}$

② $X_i + 1 - \left(\dfrac{u(x_i)}{u'(x_i)}\right)$

difficulties :

① can't use Bracketing methods ⇒ function not change sign

② " " Newton, secant " ⇒ division by zero

Bisection method :

Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

Step 2: An estimate of the root x_r is determined by

$$x_r = \frac{x_l + x_u}{2}$$

Step 3: Make the following evaluations to determine in which subinterval the root lies:

(a) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

(b) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

(c) If $f(x_l)f(x_r) = 0$, the root equals x_r; terminate the computation.

$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\% \tag{5.2}$$

where x_r^{new} is the root for the present iteration and x_r^{old} is the root from the previous iteration. The absolute value is used because we are usually concerned with the magnitude of ε_a rather than with its sign. When ε_a becomes less than a prespecified stopping criterion ε_s, the computation is terminated.

x_L	$f(x_L)$	x_u	$f(x_u)$	x_r	$f(x_r)$	$\varepsilon_a\%$

False position method :

Procedure

1. Find a pair of values of x, x_l and x_u such that $f_l=f(x_l) <0$ and $f_u=f(x_u) >0$.
2. Estimate the value of the root from the following formula (Refer to Box 5.1)

$$x_r = \frac{x_l f_u - x_u f_l}{f_u - f_l}$$

and evaluate $f(x_r)$.

3. Use the new point to replace one of the original points, keeping the two points on opposite sides of the x axis.

If $f(x_r)<0$ then $x_l=x_r$ $==>$ $f_l=f(x_r)$

If $f(x_r)>0$ then $x_u=x_r$ $==>$ $f_u=f(x_r)$

If $f(x_r)=0$ then you have found the root and need go no further!

$$\varepsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

x_L	$f(x_L)$	x_u	$f(x_u)$	x_r	$f(x_r)$	$\varepsilon_a\%$

Fixed point iteration :

Simple Fixed-point Iteration

- Rearrange the function so that x is on the left side of the equation:

$$f(x)=0 \quad \Rightarrow \quad g(x)=x$$

$$x_k = g(x_{k-1}) \qquad x_o \text{ given , k} = 1, 2, \ldots$$

- Bracketing methods are "convergent".
- Fixed-point methods may sometime "diverge", depending on the stating point (initial guess) and how the function behaves.

Conclusion

- Fixed-point iteration converges if

$$|g'(x)| \prec 1 \quad \text{(slope of the line f(x) = x)}$$

- When the method converges, the error is roughly proportional to or less than the error of the previous step, therefore it is called "linearly convergent."

x_0	$g(x_0) / x_{new}$	ε_a %

Newton Raphson method :

Step 1

Evaluate $f'(x)$ symbolically.

Step 3

Find the absolute relative approximate error $|\in_a|$ as

$$|\in_a| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100$$

x_i	$f(x_i)$	$f'(x_i)$	x_{i+1}	$\varepsilon_a \%$

Secant method :

Step 1

Calculate the next estimate of the root from two initial guesses

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Find the absolute relative approximate error

$$|\epsilon_a| = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| \times 100$$

x_{i-1}	$f(x_{i-1})$	x_i	$f(x_i)$	x_{i+1}	$\epsilon_a \%$

* function of more than one variable ⇒ Taylor series expansion

$$u_{i+1} = u_i + \frac{\partial u_i}{\partial x}(x_{1_{i+1}} - x_{1_i}) + \frac{\partial u_i}{\partial y}(y_{i+1} - y_i)$$

$$v_{i+1} = v_i + \frac{\partial v_i}{\partial x}(x_{1_{i+1}} - x_{1_i}) + \frac{\partial v_i}{\partial y}(y_{i+1} - y_i)$$

$u_{i+1}, v_{i+1} = \text{zero} \Rightarrow$ roots.

*2 linear system with two unknown ⇒

$$x_{i+1} = x_i - \frac{u_i \frac{\partial v_i}{\partial y} - v_i \frac{\partial u_i}{\partial y}}{\frac{\partial u_i}{\partial x}\frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y}\frac{\partial v_i}{\partial x}}$$

$$y_{i+1} = y_i - \frac{u_i \frac{\partial v_i}{\partial x} - v_i \frac{\partial u_i}{\partial x}}{\frac{\partial u_i}{\partial x}\frac{\partial v_i}{\partial y} - \frac{\partial u_i}{\partial y}\frac{\partial v_i}{\partial x}}$$

→ Determinant of Jacobian of system.

* linear system ⇒

① direct ② iterative ③ inverse.

Notes :

① switching 2 rows or columns does not chage the solution

② any row can multiplied by constant without chag solution

③ any row of linear mutiple of row can be added / subtractio to another row without changing solution.

* Direct → Gauss Elimination, Gauss Jordan, LU - factorization, Thomas

* Iterative → Jacobi, Gauss seidel.

① Gauss Elimination

$$\begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix} \Leftarrow \text{(our goal)}$$

echelon form

$$\underbrace{\begin{bmatrix} a & b & c \\ r & d & e \\ g & z & f \end{bmatrix}}_{\text{Numbers}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{\text{unknow}} = \underbrace{\begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}}_{\text{Number.}}$$

* we want r, g, z to be zero.

$$\begin{array}{c} R_1 \\ R_2 \\ R_3 \end{array} \left[\begin{array}{ccc|c} a & b & c & N_1 \\ r & d & e & N_2 \\ g & z & f & N_3 \end{array}\right] \xrightarrow[\text{then } R_2 - R_1 \text{ and replace new values in } R_2]{\frac{r}{a} * R_1} \text{get } r \text{ zero}$$

$$\left[\begin{array}{ccc|c} a & b & c & N_1 \\ 0 & p & u & N_4 \\ g & z & f & N_3 \end{array}\right] \xrightarrow[\text{then } R_3 - R_1 \text{ and replace new values in } R_3]{\frac{g}{a} * R_1} \text{get } g \text{ zero}$$

$$\left[\begin{array}{ccc|c} a & b & c & N_1 \\ 0 & p & u & N_4 \\ 0 & i & m & N_5 \end{array}\right] \xrightarrow[\text{then } R_3 - R_2 \text{ and replace in } R_3]{\frac{i}{p} * R_2} \text{get } i \text{ zero}$$

⇒ by doing this we finish first step of gauss Elimination Forward Elimination

⇓ then

$$\left[\begin{array}{ccc|c} a & b & c & N_1 \\ 0 & p & u & N_2 \\ 0 & 0 & t & N_6 \end{array}\right] \Longrightarrow \text{back Substitution}$$

⇓

find x_3, x_2, x_1

② gauss jordan

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Leftarrow \text{(our goal)}$$

Reduced echelon form

* Gauss Elimination pitfalls ⇒ ① Division by zero
② large Round-off error

Avoiding ⇒ by
① decrease Round off error by increase number of significant digit.
② avoid division by zero

⇓

Gaussian Elimination with Partial Pivoting.

What is Different About Partial Pivoting?

At the beginning of the k^{th} step of forward elimination, find the maximum of

$$|a_{kk}|, |a_{k+1,k}|, \ldots\ldots\ldots\ldots, |a_{nk}|$$

If the maximum of the values is $|a_{pk}|$

in the p^{th} row, $k \le p \le n$, then switch rows p and k.

Determinant of a Square Matrix Using Naïve Gauss Elimination

Example

Finding the Determinant

After forward elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

$$\begin{aligned} \det(\mathrm{A}) &= u_{11} \times u_{22} \times u_{33} \\ &= 25 \times (-4.8) \times 0.7 \\ &= -84.00 \end{aligned}$$

LU Decomposition :

$$AX = B$$

We can solve the system using LU Decomposition

Let $A = LU$ and substitute into $AX = B$.

Solve $LUX = B$ for X to solve the system.

Let $UX = Y$.

$$LY = B \text{ and } UX = Y$$

First Solve $LY = B$ for Y and then solve $UX = Y$ for X.

$$[L][U] = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

[*U*] is the same as the coefficient matrix at the end of the forward elimination step.

[*L*] is obtained using the *multipliers* that were used in the forward elimination process

Matrix Inverse by LU Decomposition

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix} \; ; \; \boxed{A = L \times U}$$

<u>Solution:-</u> First, lets find sub-matrix U.

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix} \xrightarrow[R_3 = R_3 + 1R_1]{R_2 = R_2 - 2R_1} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & -1 & 1 \end{bmatrix} \xrightarrow{R_3 = R_3 - \frac{1}{2}R_2} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix} = U$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1/2 & 1 \end{bmatrix} \; ; \; \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1/2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U}$$

$B = A^{-1}$

$A . I = A$

$A . A^{-1} = I$

$A . B = I$

$A = L . U$

$$\underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 3 \\ -1 & -3 & 0 \end{bmatrix}}_{A} . \underbrace{\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{I}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1/2 & 1 \end{bmatrix}}_{L} . \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{C} \Rightarrow \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}}_{Z} \Rightarrow \begin{bmatrix} -9 \\ 3 \\ 4 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1/2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{C} \Rightarrow \begin{bmatrix} 0 \\ 1 \\ -1/2 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 0 \\ 1 \\ -1/2 \end{bmatrix}}_{Z} \Rightarrow \begin{bmatrix} 3 \\ -1 \\ -1 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1/2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{C} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{Z} \Rightarrow \begin{bmatrix} -4 \\ 1 \\ 2 \end{bmatrix}$$

$$B = A^{-1} = \begin{bmatrix} -9 & 3 & -4 \\ 3 & -1 & 1 \\ 4 & -1 & 2 \end{bmatrix}$$

For solving z in calculator : inverse matrix L * matrix C

For solving b in calculator : inverse matrix U * matrix Z

Gauss-Seidel Method

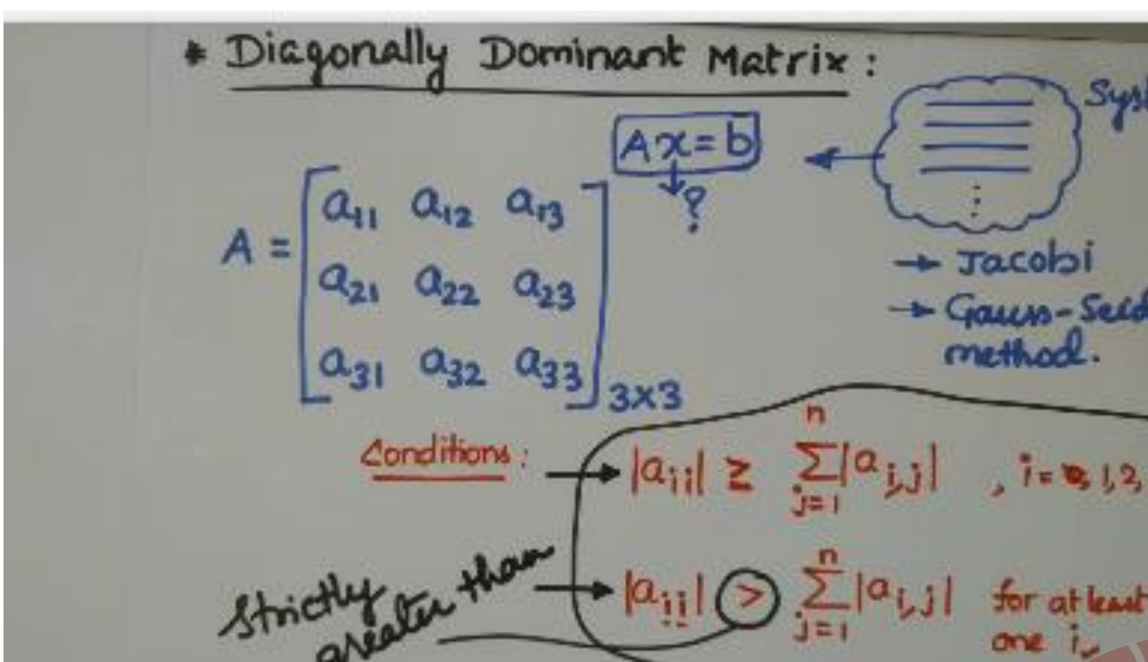
Calculate the Absolute Relative Approximate Error

$$\left|\in_a\right|_i = \left|\frac{x_i^{new} - x_i^{old}}{x_i^{new}}\right| \times 100$$

So when has the answer been found?

The iterations are stopped when the absolute relative approximate error is less than a prespecified tolerance for all unknowns.

We solve iteration in calculator



If a system of linear equations is not diagonally dominant, check to see if rearranging the equations can form a diagonally dominant matrix.

“Goodness” of our fit If

- Total sum of the squares around the mean for the dependent variable, y, is S_t $$S_t = \sum_{i=1}^{n} (y_i - y)^2$$
- Sum of the squares of residuals around the regression line is S_r $$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$
- S_t-S_r quantifies the improvement or error reduction due to describing data in terms of a straight line rather than as an average value.

$$r^2 = \frac{S_t - S_r}{S_t}$$

r^2-coefficient of determination

r=Sqrt(r^2) – correlation coefficient

- For a perfect fit

  S_r=0 and r=r^2=1, signifying that the line explains 100 percent of the variability of the data.
- For **r=r^2=0, S_r=S_t**, the fit represents no improvement.

بعد ترتيب المعادلات نحصل على:

حفظ

$$* \quad a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$* \quad b = \bar{y} - a\bar{x} \quad \begin{cases} \bar{y} = \frac{\sum y_i}{n} \\ \bar{x} = \frac{\sum x_i}{n} \end{cases}$$

او نستطيع ايجادهم من طريقة الآلة الحاسبة بإدخال المعادلات 1 و 2

* نحصل بعدها على الاقتران $F(x) = ax + b$ والذي يمثل هذه المشاهدات فلو طلب توقع نتيجة بالنسبة للاخرى نجدها من خلال الاقتران مثلاً لو طلب ايجاد $f(2)$ .. هنا تكون $(x=2)$ و y نجدها من خلال الاقتران اي اصبح بإمكاننا توقع مشاهدات اخرى

Linear Regression (special case cont.)

Does this value of a_1 correspond to a local minima or local maxima?

$$a_1 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

$$\frac{dS_r}{da_1} = \sum_{i=1}^{n} \left(-2 y_i x_i + 2 a_1 x_i^{\,2} \right)$$

$$\frac{d^2 S_r}{da_1^{\,2}} = \sum_{i=1}^{n} 2 x_i^{\,2} > 0$$

Yes, it corresponds to a local minima.

$$a_1 = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

Polynomial Model cont.

These equations in matrix form are given by

$$\begin{bmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \cdots & \left(\sum_{i=1}^{n} x_i^{m}\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^{2}\right) & \cdots & \left(\sum_{i=1}^{n} x_i^{m+1}\right) \\ \vdots & \vdots & & \vdots \\ \left(\sum_{i=1}^{n} x_i^{m}\right) & \left(\sum_{i=1}^{n} x_i^{m+1}\right) & \cdots & \left(\sum_{i=1}^{n} x_i^{2m}\right) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \vdots \\ \sum_{i=1}^{n} x_i^{m} y_i \end{bmatrix}$$

The above equations are then solved for $a_0, a_1, \ldots, a_m$

For solving in calculator : inverse matrix 1 * matrix 3 ---> matrix 2

Transformation of Data

To find the constants of many nonlinear models, it results in solving simultaneous nonlinear equations. For mathematical convenience, some of the data for such models can be transformed. For example, the data for an exponential model can be transformed.

As shown in the previous example, many chemical and physical processes are governed by the equation,

$$y = ae^{bx}$$

Taking the natural log of both sides yields,

$$\ln y = \ln a + bx$$

Let $z = \ln y$ and $a_0 = \ln a$

We now have a linear regression model where $z = a_0 + a_1 x$

(implying) $a = e^{a_o}$ with $a_1 = b$

Transformation of data cont.

Using linear model regression methods,

$$a_1 = \frac{n\sum_{i=1}^{n} x_i z_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} z_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$a_0 = \bar{z} - a_1 \bar{x}$$

Once a_o, a_1 are found, the original constants of the model are found as

$$b = a_1$$

$$a = e^{a_0}$$

Exponential Regression

→ Give $(x_i, y_i) \ldots (x_n, y_n)$

→ $F(x) = a e^{bx} = y$

$$S = \sum_{i=1}^{n} (y_i - F(x))^2$$

$\therefore \rightarrow \ln y = \ln (a e^{bx})$

$= \ln a + \ln e^{(bx)}$

$= \underbrace{\ln a}_{\text{نفرضها } a_1} + bx$

$\ln y = bX + a_1$

then solve:

$$\left.\begin{aligned} n a_1 + b \sum x &= \sum \ln y \\ a_1 \sum x + b \sum x^2 &= \sum \ln y \, x \end{aligned}\right]$$

نجد a_1 & b عن طريق الحاسبة

$a_1 = \ln a \ldots \rightarrow a = e^{a_1}$

$\therefore \rightarrow y = a e^{bx}$

Interpolants

Polynomials are the most common choice of interpolants because they are easy to:

- Evaluate
- Differentiate, and
- Integrate

Direct Method

Given 'n+1' data points (x_0,y_0), (x_1,y_1),............. (x_n,y_n), pass a polynomial of order 'n' through the data as given below:

$$y = a_0 + a_1 x + \ldots\ldots\ldots\ldots\ldots\ldots + a_n x^n .$$

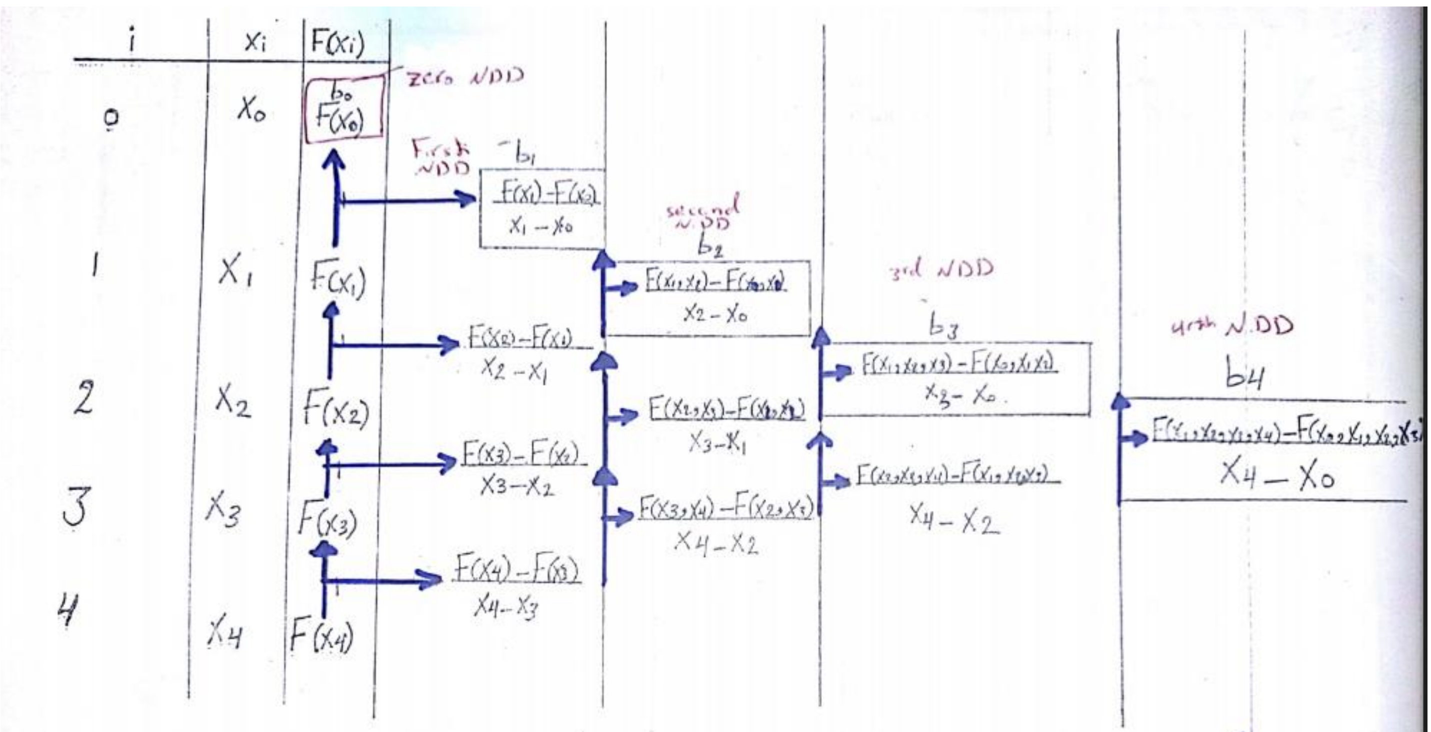
where a_0, a_1,.................. a_n are real constants.

- Set up 'n+1' equations to find 'n+1' constants.
- To find the value 'y' at a given value of 'x', simply substitute the value of 'x' in the above polynomial.

$$\# \; P_n(x) = b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + b_3(x-x_0)(x-x_1)(x-x_2) \dots$$
$$+ b_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

كله عدد المشاهدات ناقص واحد

ثم هذه هي معادلة N.D.D



* الاسهم التي تعرض تحدد بنظام جح

- General form of Lagrange poly. is :-

$$P_n(x) = \sum_{i=0}^{n} F(x_i)\, L_i(x) = f(x_0)L_0(x) + f(x_1)L_1(x) \cdots + f(x_n)L_n(x)$$

where :-

نعبر عنه بالمضروب

$$L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k}$$

Example :- Find Lagrange poly. for the following :-

1] Givin $(x_0, F(x_0))$, $(x_1, f(x_1))$ ... 2 Points

sol^n :-

x	x_0	x_1
y	$F(x_0)$	$F(x_1)$

$$P(x) = F(x_0)\, L_0(x) + F(x_1)\, L_1(x)$$

$$= F(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + F(x_1)\, \frac{x - x_0}{x_1 - x_0}$$

ملاحظة :-
Lagrange poly. of degree one ≈ the Same N.D.D poly of degree one (equal $F(x_0)$)

2] Givin $(x_0, F(x_0)), (x_1, F(x_1)), (x_2, F(x_2))$ ... 3 Points

x	x_0	x_1	x_2
y	$F(x_0)$	$F(x_1)$	$F(x_2)$

$$P_2(x) = F(x_0)\, L_0(x) + F(x_1)\, L_1(x) + F(x_2)\, L_2(x)$$

$$= F(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + F(x_1) * \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$+ F(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Numerical integration

* تذكر ان التكامل يعبر عن المساحة ضمن الاقتران

1] Trapezoidal Method: to approximate $\int_a^b F(x)\,dx$ where

$F(x) \approx P_n(x)$ ⋯> $P_n(x)$ is 1st order $(n=1)$ Lagrange interpolation

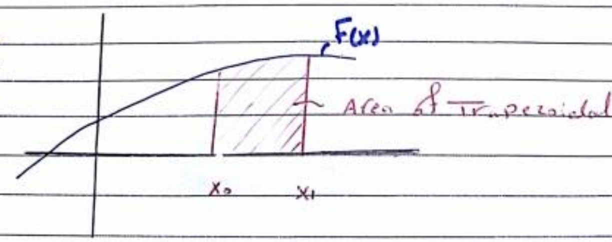
⋯> $F(x) \approx P_1(x) = F(x_0)\dfrac{x-x_1}{\underbrace{x_0-x_1}_{=-h}} + F(x_1)\dfrac{x-x_0}{\underbrace{x_1-x_0}_{=h}}$

$$F(x) = \frac{1}{h}\left[F(x_1)(x-x_0) - F(x_0)(x-x_1)\right], \quad h = x_1 - x_0$$

$$\int_a^b F(x)\,dx = \frac{1}{h}\left[F(x_1)\int_a^b (x-x_0)\,dx - F(x_0)\int_a^b (x-x_1)\right]$$

$$\int_a^b F(x)\,dx = \frac{h}{2}\left(F(x_1) + F(x_0)\right) \cdots\!> \text{Trapezoidal Rule}$$

$h = b - a$
$= x_1 - x_0$



Multiple Segment Trapezoidal Rule

The integral I can be broken into h integrals as:

$$\int_a^b f(x)dx = \int_a^{a+h} f(x)dx + \int_{a+h}^{a+2h} f(x)dx + \dots + \int_{a+(n-2)h}^{a+(n-1)h} f(x)dx + \int_{a+(n-1)h}^{b} f(x)dx$$

Applying Trapezoidal rule on each segment gives:

$$\int_a^b f(x)dx = \frac{b-a}{2n}\left[f(a) + 2\left\{\sum_{i=1}^{n-1} f(a+ih)\right\} + f(b)\right]$$

$$= \frac{h}{2}\left[f(a) + 2\left\{\sum_{i=1}^{n-1} f(a+ih)\right\} + f(b)\right]$$

Basis of Simpson's 1/3rd Rule

Substituting values of a_0, a_1, a_2 give

$$\int_a^b f_2(x)dx = \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Since for **Simpson's 1/3rd Rule**, the interval **[a, b]** is broken into **2 segments**, the segment width:

$$h = \frac{b-a}{2}$$

Multiple Segment Simpson's 1/3rd Rule

$$\int_a^b f(x)dx = \frac{h}{3}[f(x_0) + 4\{f(x_1) + f(x_3) + \ldots + f(x_{n-1})\} + \ldots]$$

$$\ldots + 2\{f(x_2) + f(x_4) + \ldots + f(x_{n-2})\} + f(x_n)\}]$$

$$= \frac{h}{3}\left[f(x_0) + 4\sum_{\substack{i=1 \\ i=odd}}^{n-1} f(x_i) + 2\sum_{\substack{i=2 \\ i=even}}^{n-2} f(x_i) + f(x_n)\right]$$

$$= \frac{b-a}{3n}\left[f(x_0) + 4\sum_{\substack{i=1 \\ i=odd}}^{n-1} f(x_i) + 2\sum_{\substack{i=2 \\ i=even}}^{n-2} f(x_i) + f(x_n)\right]$$

Initial Value Problem

فكرة هذا الجزء [illegible]

I] Euler's Method

For the equation $\frac{dy}{dt} = F(t, y)$, $0 \leq t \leq b$, $y(t_0) = x$

* Consider a unifor partition , then from tylor poly. we have :

$\overset{a}{t_0}, t_1, t_2 \dots \overset{b}{t_n}$

$y(t) = y(t_i) + y'(t_i)(t - t_i)$

at Point $t+1$: $y(t+1) = y(t_i + h) = y(t_i) + y'(t_i)h$
$= y(t_i) + F(t_i, y_i)h$

$w_0 = y_0 = y(a)$ where, $h = \frac{b-a}{n}$ عدد النقاط

$w_{i+1} = w_i + h\,F(t_i, y_i)$ $w_i = y_i = y(t_i)$

Improvements of Euler's method

- A fundamental source of **error** in Euler's method is that the **derivative at the beginning of the interval is assumed to apply across the entire interval**.
- Two simple modifications are available to circumvent this shortcoming:
  - Heun's Method
  - The Midpoint (or Improved Polygon) Method

Runge-Kutta 2nd Order Method

For $\frac{dy}{dx} = f(x, y), \quad y(0) = y_0$

Runge Kutta 2nd order method is given by

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

where

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h)$$

Heun's Method

Heun's method

Here $a_2 = 1/2$ is chosen

$$a_1 = \frac{1}{2}$$

$$p_1 = 1$$

$$q_{11} = 1$$

resulting in

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h$$

where

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_{i+1}, y_{i+1}) = f(x_i + h, y_i + k_1 h) = f(x_i + h, y_i + f(x_i, y_i)h)$$

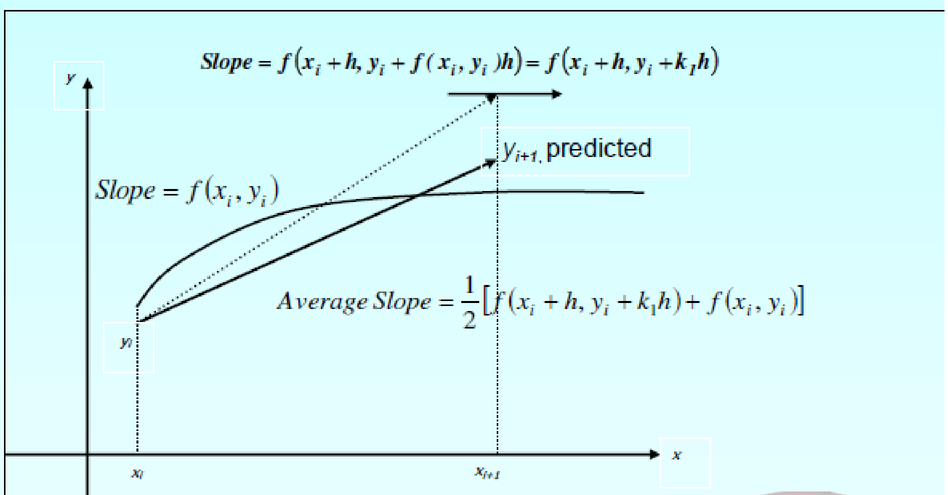


Figure 1 Runge-Kutta 2nd order method (Heun's method)

Runge-Kutta 4th Order Method

For $\frac{dy}{dx} = f(x, y), y(0) = y_0$

Runge Kutta 4th order method is given by

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

where

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

$$k_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right)$$

$$k_4 = f(x_i + h, y_i + k_3 h)$$