ch 2

chap 2.1

3.3

- Fixed point . $P = g(P)$

- Fixed point Iteration theorem :

Assume $g \in C[a,b]$.

• If $g(x) \in [a,b]$ for all $x \in [a,b]$, then g has a Fixed point in $[a,b]$.

and • If $|\bar{g}(x)| \leq K < 1$ for all $x \in (a,b)$, then g has a unique Fixed point in $[a,b]$.

- Fixed point Iteration theorem :

Assume the following $g, \bar{g} \in C[a,b]$, K is constant positive , $P_0 \in [a,b]$ is the starting point $g(x) \in [a,b]$ for all $x \in [a,b]$.

• If $|\bar{g}(x)| \leq K < 1$ for all $x \in [a,b]$, then the FPI converge to the unique FP, P

(P is called attractive Fixed point)

• If $|\bar{g}(x)| > 1$ for all $x \in [a,b]$, the the FPI diverge.

(P is called repelling Fixed point).

- Corollary :

Assume g satisfies conditions of FPI Theorem 2↑ . If P_n is used to approximate the Fixed point P, then an upper bounds for the error are :

A. $|P - P_n| \leq K^n |P - P_0|$ for all $n \geq 1$.

B. $|P - P_n| \leq \frac{K^n}{1-K} |P_1 - P_0|$ for all $n \geq 1$.

اذا كانت P موجودة

اذا كانت P غير موجودة

★ K upper bound of $|\bar{g}(x)|$.

لـ كوان يوجد عينة n حسب المعادلة التالية ..

★ Bisection Method of Bolzano:

- this method use to solve $f(x)=0$, and depends on Bolzano theorem.
- conditions required: $f \in C[a,b]$ and $f(a)f(b) < 0$.
- First iteration $c_0 = \frac{a_0 + b_0}{2}$ (midpoint).
- Now find $f(c_0)$, we have three cases:
  - if $f(c_0) = 0$, then c_0 is the root and we are done.
  - if $f(c_0)f(a_0) < 0$, then $[a_1, b_1] = [a_0, c_0]$ and $c_1 = \frac{a_1 + b_1}{2} = \frac{a_0 + b_0}{2}$.
  - if $f(c_0)f(b_0) < 0$, then $[a_1, b_1] = [c_0, b_0]$ and $c_1 = \frac{a_1 + b_1}{2} = \frac{c_0 + b_0}{2}$.

لا ارسم خط الاتجاه وأوجد الاشارة ويصير أوخذ الفترة اللي بين فيها + و −

★ In general, $c_n = \frac{a_n + b_n}{2}$ is Bisection method.

★ Bisection Theorem:

- Assume that $f \in C[a,b]$, $\exists$ a number $r \in [a,b]$ s.t $f(r)=0$, $f(a)f(b) < 0$ and $\{c_n\}_{n=0}^{\infty}$ represents the sequence of midpoints generated by the bisection method.
- Then an upper bound of the error is $|r - c_n| \leq \frac{b-a}{2^{n+1}}$.
- The sequence $\{c_n\}_{n=0}^{\infty}$ converges to the zero r.

★ Remark:

The Bisection theorem provides a starting to find the number of iteration for a given accuracy δ:

$$\frac{b-a}{2^{n+1}} < \delta \longrightarrow n > \frac{\ln\left(\frac{b-a}{\delta}\right)}{\ln 2} - 1.$$

★ False position Method. (converges faster)

- conditions to solve $f(x)=0$: $f \in C[a,b]$ and $f(a)f(b)<0$.
- As in Bisection method, we have three cases
  - If $f(a_0)f(c_0)<0$, then the zero $r \in [a_0, c_0]$.
  - If $f(c_0)f(b_0)<0$, then the zero $r \in [c_0, b_0]$
  - If $f(c_0)=0$, then the zero $r = c_0$.

In general : $$c_n = b_n - \left(\frac{b_n - a_n}{f(b_n)-f(a_n)}\right) f(b_n).$$

★ Newton's method iteration : $p_{n+1} = p_n - \frac{f(p_n)}{\bar{f}(p_n)}$, $n = 0, 1, 2, \ldots$ (Faster than Bisection and false position method)

★ secant Method : $p_{n+2} = p_{n+1} - \left(\frac{p_{n+1} - p_n}{f(p_{n+1}) - f(p_n)}\right) f(p_{n+1})$. (as fast as Newton method)

p_0, p_1 : given.

★ Multiplicity of Roots :

- we say $f(x) = 0$ has a root of order M at $x = p$ (p has multiplicity M) iff

$f(p) = 0$, $\bar{f}(p) = 0$, $\bar{\bar{f}}(p) = 0$, ... , $f^{(M-1)}(p) = 0$, $f^{M}(p) \neq 0$.

- Lemma : If $f(x) = 0$ has a root p and $\exists$ a continuous function $h(x)$ s.t

$f(x) = (x-p)^M h(x)$ where $h(p) \neq 0$. then the root p has multiplicity M.

★ Speed of convergence : for any Method.

- If there exist two positive constants $A \neq 0$ and $R > 0$ s.t $\lim_{n\to\infty} \frac{|p - p_{n+1}|}{|p - p_n|^R} = \lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^R} = A$

then we say that $\{p_n\}$ converges to p with order of convergence R.

- We use the order of convergence R to measure the speed of convergence of any Method :

- If $R = 1$ then the convergence of $\{p_n\}$ is linear
- If $R = \frac{3}{2}$ then the convergence of $\{p_n\}$ is super linear
- If $R = 2$ then the convergence of $\{p_n\}$ is quadratic
- If $R = 3$ then the convergence of $\{p_n\}$ is cubic.

- When $R \uparrow \rightarrow$ speed $\uparrow \rightarrow$ error $\downarrow$.

- A is called the asymptotic error constant.

★ Speed of Newton's Method.

Assume Newton iteration $p_{n+1} = p_n - \frac{f(p_n)}{\bar{f}(p_n)}$, produces a sequence $\{p_n\}$ that converges to the root p of the function $f(x)$. Then:

1. If p is simple root ($M=1$), then Newton iteration $\{p_n\}$ converges to p quadratically ($R=2$) with $A = \left|\frac{\bar{\bar{f}}(p)}{2\bar{f}(p)}\right|$ and $\lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^2} = A$.

2. If p is a multiple root (of order $M>1$) then Newton's iteration $\{p_n\}$ converges to p linearly ($R=1$) with $A = \frac{M-1}{M}$ and $\lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|} = A$.

★ Accelerated Newton-Raphson iteration.

- Assume Newton-Raphson iteration $p_{n+1} = p_n - \frac{f(p_n)}{\bar{f}(p_n)}$, produces a sequence $\{p_n\}$ that converges to the root p.

- Assume p is a multiple root (of order $M>1$). Then

1. Newton's iteration $\{p_n\}$ converges linearly ($R=1$) with $A = \frac{M-1}{M}$ and $\lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|} = A$.

2. The Modification of Newton's iteration $p_{n+1} = p_n - \frac{M f(p_n)}{\bar{f}(p_n)}$, given p_0, $n = 0, 1, 2, \ldots$ converges quadratically ($R=2$) to p and $A = \lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^2}$.

★ speed of convergence for secant Method:

$$p_{n+2} = p_{n+1} - \left(\frac{p_{n+1} - p_n}{f(p_{n+1}) - f(p_n)}\right) f(p_{n+1}).$$

1. If p is simple root ($M=1$), then secant Method iteration $\{p_n\}$ converges to p with $R = 1.618$ and $A = \lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^{1.618}} = \left|\frac{\bar{\bar{f}}(p)}{2\bar{f}(p)}\right|^{0.618}$

2. If p is multiple root ($M>1$), then secant Method iteration converges to p with $R=1$ and A depends on $f(x)$.

★ speed of convergence for Bisection Method :

$$R = 1 \quad \text{and} \quad A = \frac{1}{2} \longrightarrow \frac{|E_{n+1}|}{|E_n|} \simeq \frac{1}{2}$$

★ speed of convergence for False position Methods :

$R = 1$, A depends on $f(x)$, $\frac{|E_{n+1}|}{|E_n|} \approx A$.

★ speed of convergence for Fixed point Iteration.

$$R = M \quad , \quad A = \left| \frac{g^{(K)}(P)}{K!} \right| .$$

Chapter 3:

3.3 : Solving nxn linear and nonlinear systems.

- Linear system : $AX = b$

- Cost (complexity) : is the number of operations $+, -, \times, \div$ required to complete a certain calculation

- The cost of finding $|A_{nxn}|$ for $n \geq 2$ is :

1. Cost $= n! \sum_{k=2}^{n} \frac{2k-1}{k!}$ or

2. Cost $= [n!e - 2]$ where $[\text{greatest integer}]$ is the greatest integer functions and $e \approx 2.718$.

- If A is nxn matrix, then the cost of calculating A^2 is $2n^3 - n^2$.

- Back word substitution :

used to solve a linear system of equations that has an upper-triangle coefficeint matrix.

* Cost of Backword substitution $= n^2$

- Cost of Gaussian Elimination - GE

1. The cost of reducing nxn linear system to an upper-triangle system using G.E is

$$\frac{4n^3 + 3n^2 - 7n}{6}$$

upper $\frac{n^3}{3}$ system

2. The cost of solving nxn linear system using G.E is $\frac{4n^3 + 9n^2 - 7n}{6}$ (total cost)

Gaussian Elimination and pivoting :

→ is two types :

1. Trival pivoting : - If $a_{pp} \neq 0$, then do not switch rows.
   - If $a_{pp} = 0$, then switch row P by the first row K below P s.t $a_{kp} \neq 0$.

2. Partial pivoting : - If $a_{pp} \neq 0$ or $a_{pp} = 0$, choose the pivotal - $|a_{kp}| = \max \{ |a_{pp}|, |a_{p+1,p}|, \dots |a_{np}| \}$.
   - reducing the error being propagated when using a limit digit arithmatic.

LU- Factorization :

To solve the linear system $AX = b$:

- write $A = LU$ where L is lower triangle matrix. and U is upper triangle matrix.
- Let $y = UX \Rightarrow AX = b$ becomes $LUX = b \rightarrow Ly = b$.
- Now solve $Ly = b$ by F.S and Find y.
- Then solve $UX = y$ by B.S and Find x.

$\Rightarrow$ Cost of LU = cost of G.E

Remark : The total cost of solving any linear $n \times n$ system $Ax = b$ using LU-Factorization.

$$= \text{Cost of } LU + \text{Cost of F.S} + \text{Cost of B.S}$$

$$= \frac{4n^3 - 3n^2 - n}{6} + n^2 - n + n^2$$

($\frac{4n^3 - 3n^2 - n}{6}$: Cost of $A = LU$; $n^2 - n$: cost of F.S ($Ly = b$) ; n^2 : Cost on B.S)

$$= \frac{4n^3 + 9n^2 - 7n}{6}$$

$$= \text{Cost of G.E}$$

★ Cramer's Rule .

→ To solve the linear $n \times n$ system $Ax = b$:

- Find $|A|$, $|A| \neq 0$.

- The solution $x = \begin{pmatrix} x_1 \\ x_2 \\ x_n \end{pmatrix}$ is obtained as follow :

$x_i = \frac{|A_i|}{|A|}$ where A_i is obtained by replacing the i^{th} column of A by column b for all $i = 1, 2, 3, \ldots$.

→ Cost of Cramer Rule :

$(n+1) D_n + n$, where D_n is the cost of $|A|$. في الة حاسبة قبل

★ Solving nonlinear systems :

1 Newton's Method .

- Given 2×2 nonlinear system

$f(x,y) = 0$, $g(x,y) = 0$ with initial point (x_0, y_0) .

- This method find a sequence of points (x_1, y_1) , (x_2, y_2) , ... that approximate (x, y) .

- we will only find (x_1, y_1) as follow : $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - J^{-1}(x_0, y_0) \begin{pmatrix} f(x_0, y_0) \\ g(x_0, y_0) \end{pmatrix}$, where J is the Jacobian matrix given by

$J(x,y) = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}$. $\rightarrow$ $J^{-1}(x,y) = \frac{1}{f_x g_y - f_y g_x} \begin{pmatrix} g_y & -f_y \\ -g_x & f_x \end{pmatrix}$.

To find $(x_2, y_2) = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - J^{-1}(x_1, y_1) \begin{pmatrix} f(x_1, y_1) \\ g(x_1, y_1) \end{pmatrix}$.

[2] Fixed point iteration (F.P.I).

- This method can be used to solve 2×2 or 3×3 nonlinear systems.

- For 2×2 nonlinear system:

- write $x = g_1(x, y)$.
  $y = g_2(x, y)$ ... ①

- Given $(x_0, y_0) = (p_0, q_0)$.

- The FPI is $p_{n+1} = g_1(p_n, q_n)$, $q_{n+1} = g_2(p_n, q_n)$.

- For 3×3 nonlinear system:

- write $x = g_1(x, y, z)$
  $y = g_2(x, y, z)$ ... ②
  $z = g_3(x, y, z)$.

- Given $(x_0, y_0, z_0) = (p_0, q_0, r_0)$.

- The FPI is $p_{n+1} = g_1(p_n, q_n, r_n)$, $q_{n+1} = g_2(p_n, q_n, r_n)$, $r_{n+1} = g_3(p_n, q_n, r_n)$.

Def:

- The point (p, q) is fixed point of the system ① if $p = g_1(p, q)$ and $q = g_2(p, q)$.

- The point (p, q, r) is fixed point of the system if $p = g_1(p, q, r)$ and $q = g_2(p, q, r)$ and $r = g_3(p, q,$

: Convergence of FPI - Two dimensions.

- Assume (P, Q) is fixed point of $x = g_1(x,y)$ and $y = g_2(x,y)$.
- If (P_0, Q_0) is sufficiently close to (P, Q) and if
  A. $|g_{1x}(P,Q)| + |g_{1y}(P,Q)| < 1$ and
  B. $|g_{2x}(P,Q)| + |g_{2y}(P,Q)| < 1$, Then the FPI convergence to the fixed point (P, Q).

Remark :

1. Convergence of FPI for three dimensions follows similarly to Theorem above by adding z-component.
2. إذا A, B ما تحقق ما يقدر أحكم conv أو Div.

3 Seidel Iteration (improvement of FPI).

- This method can be used to solve 2x2 or 3x3 nonlinear system.
- For 2x2 nonlinear system :
  - write $x = g_1(x,y)$, $y = g_2(x,y)$.
  - Given $(x_0, y_0) = (P_0, Q_0)$.
  - The seidel Iteration is $P_{n+1} = g_1(P_n, Q_n)$, $Q_{n+1} = g_2(P_{n+1}, Q_n)$.
- For 3x3 nonlinear system :
  - write $x = g_1(x,y,z)$, $y = g_2(x,y,z)$, $z = g_3(x,y,z)$
  - Given $(x_0, y_0, z_0) = (P_0, Q_0, r_0)$.
  - The seidel Iteration is

$$P_{n+1} = g_1(P_n, Q_n, r_n)$$

$$Q_{n+1} = g_2(P_{n+1}, Q_n, r_n)$$

$$r_{n+1} = g_3(P_{n+1}, Q_{n+1}, r_n).$$

Chapter 4:

- Interpolations means polynomial approximation.
  - given a function $f(x)$ on $[a,b] = [x_0, x_n]$.
  - given $n+1$ points $(x_0, \overset{f(x_0)}{y_0}), (x_1, \overset{f(x_1)}{y_1}), \dots, (x_n, \overset{f(x_n)}{y_n})$ through the partition
    $$a = x_0 < x_1 < x_2 < \cdots < x_n = b.$$
  - We need approximate $f(x)$ by a polynomial of order at most n passing through these $n+1$ points ("nodes") on $[a,b]$.
  - That is, $f(x) \approx \underbrace{P_n(x)}_{\text{is called interpolation polynomial}} + \underbrace{E_n(x)}_{\text{truncation error}}$ on $[a,b]$.

Th. Given $n+1$ points: $(x_0, y_0), (x_1, y_1), \dots (x_n, y_n)$, Then $\exists$ a unique polynomial $P_n(x)$ with degree $\leq n$ that passes through these point.

Remark: If $f(x)$ is given and analytic at x_0 ("has continuous derivatives of all derivatives of all orders and can be represented as Taylor series in an interval about x_0"), then we can use Taylor polynomial Approximation to estimate $f(x)$ by a Taylor polynomial.

Theorem : Taylor polynomial Approximation :

- Assume $f \in C^{n+1}[a,b]$ and $x_0 \in [a,b]$ is fixed.
- If $x \in [a,b]$, then $f(x) \simeq P_n(x) + E_n(x)$ where $P_n(x)$ is the Taylor polynomial of degree n that estimates $f(x)$ on $[a,b]$ given by $P_n(x) = \sum_{k=0}^{n} \frac{f^{(n)}(x_0)}{k!} (x-x_0)^k$.

and $E_n(x)$ is the truncation error given by $E_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x-x_0)^{n+1}$.

→ However, f is usually not known or hard to compute. So How to find $P_n(x)$?

1. Lagrange Interpolation : $P_n(x)$ is the Lagrange polynomial.
2. Newton Interpolation : $P_n(x)$ is the Newton polynomial.

I Lagrange polynomial :

- Linear interpolation uses a line segment that passes through two points.

- In general, the Lagrange polynomial of degree at most n passes through $n+1$ points : $(x_0, y_0), (x_1, y_1), \dots (x_n, y_n)$ has the form $P_n(x) = \sum_{k=0}^{n} y_k \, L_{n,k}(x)$.

$$L_{n,k} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}.$$

Def : uniform partition :

- The partition of $[a,b] = [x_0, x_n]$ is **uniform** if the nodes $x_0, x_1, x_2, \dots x_n$ are equally spaced.

- That is, $x_k = x_0 + hk$ for $k = 0, 1, 2, \dots n$.

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

[2] Newtons Polynomial :

Th (Newton polynomial):

- Given $x_0, x_1, x_2, \dots x_n$, $n+1$ distinct numbers in $[a,b]$.
- Then, $\exists$ a unique polynomial $P_n(x)$ « called Newton Polynomial » of degree at most n s.t $f(x_i) = P(x_i)$ for $i = 1, 2, \dots n$.
- Furthermore, Newton polynomial is given by

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) + \dots a_n(x-x_0)(x-x_1)\dots(x-x_{n-1}).$$

Where the coefficients of Newton's polynomial are given by the divided difference

$$a_k = f[x_0, x_1, \dots, x_k] \text{ for } k = 0, 1, \dots n.$$

⟶ That is :

$a_0 = F[x_0] = f(x_0) = y_0$: Zero divided differences.

$$a_1 = F[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \quad : \text{First divided differences.}$$

$$a_2 = F[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \quad : \text{Second divided differences.}$$

★ Divided Difference Table for $y = f(x)$:

x_k	$f(x_k) = y_k$	1st D.D	2nd D.D	3rd D.D
x_0	$y_0 = a_0$			
x_1	y_1	$f[x_0, x_1] = a_1$		
x_2	y_2	$f[x_1, x_2]$	$f[x_0, x_1, x_2] = a_2$	
x_3	y_3	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3] = a_3$
x_4	y_4	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$

Error terms and error bounds.

* Error terms $E_n(x)$:

$$E_n(x) = f(x_k) - P_n(x_k) = y_k - y_k = 0.$$

- Theorem ($E_n(x)$):
- Assume $f \in C^{n+1}[a,b]$ and $x_0, x_1, \dots x_n \in [a,b]$ are $n+1$ nodes.
- Then $f(x) = P_n(x) + E_n(x)$ where $E_n(x)$ is the error term given by

$$E_n(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_n) f^{(n+1)}(c)}{(n+1)!}$$ for some $c = c(x)$ lies in $[a,b]$.

→ $P_n(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + \cdots + y_n L_{n,n}(x)$. (Lagrang polynomial).

→ $P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$. (Newton polynomial)

- How to find an upper bound for the error term $E_n(x)$?
- That is, we need to find some constant s.t $|E_n(x)| \le$ constant.

- Finding the upper bound depends on whether the nodes are equally spaced (uniform partition) or not (not uniform partition).

* Th (upper bound of the error term for Interpolation - uniform partition).

$n=1$ ① Given (x_0, y_0), (x_1, y_1), $(n=1)$ with $E_1(x) = \frac{(x-x_0)(x-x_1)\bar{f}(c)}{2!}$, Then

$|E_1(x)| \le \frac{h^2 M_2}{8}$ where $M_2 = \max|\bar{f}(x)|$. $h = x_1 - x_0$

$n=2$ ② Given (x_0, y_0), (x_1, y_1), (x_2, y_2) with $E_2(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\bar{\bar{f}}(c)}{3!}$ Then

$|E_2(x)| \le \frac{h^3 M_3}{9\sqrt{3}}$ where $M_3 = \max|\bar{\bar{f}}(x)|$. $h = \frac{x_2 - x_0}{2}$

$n=3$ ③ Given (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) with $E_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3) f^{(4)}(c)}{4!}$

Then $|E_3(x)| \le \frac{h^4 M_4}{24}$ where $M_4 = \max|f^{(4)}(x)|$. $h = \frac{x_3 - x_0}{3}$

curve fitting.

★ upper bound of the error term for interpolation - Non uniform partition.

- Given (x_0, y_0), (x_1, y_1), (x_2, y_2) with $E_2(x) = \frac{(x-x_0)(x-x_1)(x-x_2)\,\tilde{f}(c)}{3!}$ Then

$$|E_2(x)| \leq \frac{|\phi_2(x)|\,|\tilde{f}(c)|}{6}$$ where:

$|\phi_2(x)|$ is an upper bound of $\phi_2(x) = (x-x_0)(x-x_1)(x-x_2)$ and

$|\tilde{f}(c)|$ is an upper bound of $|\tilde{f}(c)|$.

chapter 4 done

Chapter 5 : curve fitting.

- To handle the errors, we use norms to measure how far the curve $y=f(x)$ lies from the data.

$|e_k| = |f(x_k) - y_k|$, $K = 1, 2, \ldots, n$.

- we consider the following norms :

1. Maximum error : $E_\infty(f) = \max |e_k|$, $k \in \{1, 2, \ldots n\}$.
2. Average error : $E_1(f) = \frac{1}{n} \sum_{n=1}^{n} |e_k|$.
3. Root - mean - square error : $E_2(f) = \sqrt{\frac{1}{n} \sum_{k=1}^{n} |e_k|^2}$.

★ Finding the Least - squares line :

- Given n distinct points : $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$.
- The least square line $y = f(x) = AX + B$ is the line that minimizes the RMSE , $E_2(f)$

$$E_2(f) = \sqrt{\frac{\sum (f(x_i) - y_i)^2}{n}} \Rightarrow n E_2^2 = \sum_{i=1}^{n} (f(x_i) - y_i)^2 .$$

- E_2 is minimized iff $E(A,B) = \sum_{i=1}^{n} (Ax_i + B - y_i)^2$ is minimized.

$$\rightarrow \frac{\partial E}{\partial A} = 2 \sum_{i=1}^{n} (Ax_i + B - y_i)\, x_i = 0 \iff \sum_{i=1}^{n} (Ax_i^2 + Bx_i - y_i x_i) = 0 .$$

$$A \sum_{i=1}^{n} x_i^2 + B \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \quad \ldots \ (1)$$

$$\rightarrow \frac{\partial E}{\partial B} = 2 \sum_{i=1}^{n} (Ax_i + B - y_i) = 0 \iff \sum_{i=1}^{n} (Ax_i + B - y_i) = 0$$

$$A \sum_{i=1}^{n} x_i + nB = \sum_{i=1}^{n} y_i \quad \ldots \ (2)$$

equation (1) and (2) are called the linear normal equations and used to find the coefficients A and B.

5.3 → هذا الدفتر ما بتخلص من

Chapter 6: Numerical Differentiation.

★ Notation:

- $f_K = f(x_0 + Kh)$, $K = 0, \pm 1, \pm 2, \pm 3$.
- That is, $f_0 = f(x_0)$

$f_1 = f(x_0 + h)$

$f_{-1} = f(x_0 - h)$ . . .

$f_2 = f(x_0 + 2h)$.

★ Deference Formula for $\bar{f}(x_0)$

1. C.D.F of order $o(h^2)$: $\bar{f}(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h} + Error(Gh)$

$$\bar{f}(x_0) \approx \frac{f_1 - f_{-1}}{2h} + \frac{-h^2 \bar{\bar{f}}(c)}{6}$$

2. F.D.F of order $o(h^2)$: $\bar{f}(x_0) \approx \frac{-3f_0 + 4f_1 - f_2}{2h} + \frac{h^2 \bar{\bar{f}}(c)}{3}$.

3. B.D.F of order $o(h^2)$: $\bar{f}(x_0) \approx \frac{3f_0 - 4f_{-1} + f_{-2}}{2h} + \frac{h^2 \bar{\bar{f}}(c)}{3}$

4. C.D.F of order $o(h^4)$: $\bar{f}(x_0) \approx \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + \frac{h^4 f^{(5)}(c)}{30}$.

★ Deference Formula for $\bar{\bar{f}}(x_0)$:

1. C.D.F of order $o(h^2)$: $\bar{\bar{f}}(x_0) \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2} + \frac{-h^2 f^{(4)}(c)}{12}$.

2. F.D.F of order $o(h^2)$: $\bar{\bar{f}}(x_0) \approx \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} + \frac{11 h^2 f^{(4)}(c)}{12}$.

3. B.D.F of order $o(h^2)$: $\bar{\bar{f}}(x_0) \approx \frac{2f_0 - 5f_{-1} + 4f_{-2} - f_{-3}}{h^2} + \frac{11 h^2 f^{(4)}(c)}{12}$.

4. C.D.F of order $o(h^4)$: $\bar{\bar{f}}(x_0) \approx \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + \frac{h^4 f^{(6)}(c)}{90}$

6.2 : Derivation of D.F's .

- Recall Taylor ~~self~~ expansion of $f(x)$ about x_0 :

$$f(x) = f(x_0) + \bar{f}(x_0)(x-x_0) + \frac{\bar{\bar{f}}(x_0)}{2!}(x-x_0)^2 + \frac{\bar{\bar{\bar{f}}}(x_0)}{3!}(x-x_0)^3 + \dots$$

- clearly $\longrightarrow$ when $x = x_0 + h \longrightarrow x - x_0 = h$:

$$f_1 = f(x_0+h) = f(x_0) + h\bar{f}(x_0) + \frac{h^2}{2!}\bar{\bar{f}}(x_0) + \frac{h^3}{3!}\bar{\bar{\bar{f}}}(x_0) + \dots$$

$$f_{-1} = f(x_0-h) = f(x_0) - h\bar{f}(x_0) + \frac{h^2}{2!}\bar{\bar{f}}(x_0) - \frac{h^3}{3!}\bar{\bar{\bar{f}}}(x_0) + \dots$$

$$f_2 = f(x_0+2h) = f(x_0) + 2h\bar{f}(x_0) + \frac{(2h)^2}{2!}\bar{\bar{f}}(x_0) + \frac{(2h)^3}{3!}\bar{\bar{\bar{f}}}(x_0) + \dots$$

$$f_{-2} = f(x_0-2h) = f(x_0) - 2h\bar{f}(x_0) + \frac{(2h)^2}{2!}\bar{\bar{f}}(x_0) - \frac{(2h)^3}{3!}\bar{\bar{\bar{f}}}(x_0) + \dots$$

$$f_k = f(x_0+kh) = f_0 + kh\bar{f}(x_0) + \frac{(kh)^2}{2!}\bar{\bar{f}}(x_0) + \frac{(kh)^3}{3!}\bar{\bar{\bar{f}}}(x_0) + \dots$$

ch.6 done.

الدفتر تم

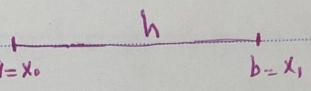
Ch 7 :

7.1 : Numarical Integration :

☆ closed Newton -- Cotes quadrature Formula :

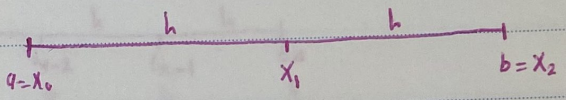
Th : Assume that $x_k = x_0 + kh$ are equally spaced nodes with $f_k = f(x_k)$, Then

[a] Trapezoidal Rule is :



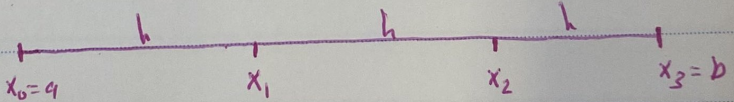
$$\int_a^b f(x)\,dx = \int_{x_0}^{x_1} f(x)\,dx \approx \underbrace{\frac{h}{2}(f_0 + f_1)}_{Q[F]} \quad \text{with error} \quad \underbrace{\frac{-h^3 f''(c)}{12}}_{E(f)}$$

[b] simpson's $\frac{1}{3}$ Rule is :



$$\int_a^b f(x)\,dx = \int_{x_0}^{x_2} f(x)\,dx \approx \frac{h}{3}(f_0 + 4f_1 + f_2) \quad \text{with error} \quad \frac{-h^5 f^{(4)}(c)}{90}$$

[c] simpson's $\frac{3}{8}$ Rule is :



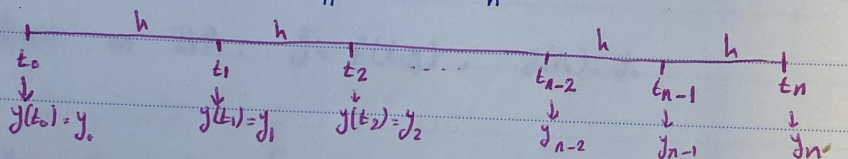
$$\int_a^b f(x)\,dx = \int_{x_0}^{x_3} f(x)\,d(x) \approx \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) \quad \text{with error} \quad \frac{-3h^5 f^{(4)}(c)}{80}$$

Ch 9: Numarical Approximation for IVP :

- Given the IVP : $\frac{dy}{dt} = f(t, y(t))$ with $y(t_0) = y_0$
- To estimate the solution $y(t)$ numerically :
  - we find the values of y at different values of t.
  - Then approximate y using interpolation.

★ General principle :

- Given an IVP : $\dot{y} = f(t, y)$, $y(t_0) = y_0$.
- To estimate the values of y on $[a, b] = [t_0, t_n]$:
  - → Find $h = \frac{t_n - t_0}{n} = \frac{b-a}{n}$.



  - → Find $y(t_k) = y_k$, where $k = 0, 1, \dots n$ and $t_k = t_0 + kh$.

- to find $y_1, y_2, \dots$.

II Euler's Method :

- given an IVP : $\dot{y} = f(t, y)$, $y(t_0) = y_0$
- $h = \frac{t_n - t_0}{n} = \frac{b-a}{n}$
- $y_1 = y_0 + h f(t_0, y_0)$.
- $y_2 = y_1 + h f(t_1, y_1)$.

in general : $y_n = y_{n-1} + h f(t_{n-1}, y_{n-1})$.

$\leq \frac{h}{1-K} |t_1 - t_0|$

[2] Taylor's Method of order 2 :

- Given an IVP : $\bar{y} = f(t,y)$, $y(t_0) = y_0$ with $[a,b] = [t_0, t_n]$, $h = \frac{b-a}{n}$
- $y_1 = y_0 + h f(t_0, y_0) + \frac{h^2}{2} \bar{f}(t_0, y_0)$.
- in general : $y_{k+1} = y_k + h f(t_k, y_k) + \frac{h^2}{2} \bar{f}(t_k, y_k)$, $k = 0, 1, 2, \dots n-1$.

[3] Heun's Method :

- Given an IVP : $\bar{y} = f(t,y)$, $y(t_0) = y_0$
- $y_1 = y_0 + \frac{h}{2} \left[f(t_0, y_0) + f(t_1, y_0 + h f(t_0, y_0)) \right]$.
- $y_2 = y_1 + \frac{h}{2} \left[f(t_1, y_1) + f(t_2, y_1 + h f(t_1, y_1)) \right]$.

ch. 9 done.