Lecture - 1

Numerical Analysis: Study of various methods to solve
→ differential equations
→ systems of linear equations
→ $f(x) = \alpha$

and approximate functions, study corresponding errors.
↳ Constructing approximate solu^n's by hand.

Ex: $e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$

Approximating e :- value of the exponential function exp(x) or e^x at the point $x = 1$ is e.

→ use Taylor's theorem to approximate it.

$$f(x) = f(a) + f'(a)(x-a) + \dots + \frac{f^{(k)}(a)}{k!}(x-a)^k$$

$$+ \underbrace{\frac{f^{(k+1)}(c)}{(k+1)!}(x-a)^{k+1}}_{\text{Error}} \; ; \; c \in (0,1), c \in \mathbb{R}$$

where, $x < c < a$.

using above expression,

$$e = \exp(1) = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!} + \underbrace{\frac{e^c}{(n+1)!}}_{\text{error at } n^{th} \text{ approxim}}$$

$\forall c \in (0,1)$ $e^c < 3$, ∴ can compute n where $\frac{e^c}{(n+1)!} <$ prescribed error.

ex: if $\frac{e^c}{(n+1)!} < 10^{-10}$, find n s.t $\frac{3}{(n+1)!} < 10^{-10}$

⇒ $n \geq 13$ for first 10 decimal points.

Can use taylor's polynomial to approximate any e^α to given accuracy.

Page No:
Date:

Aspects of Numerical Analysis:
→ theory behind calculation.
→ Computation (by calculators).

Representing a number: all numbers have different representations in different bases

Lecture 2

The error that is produced when a calculator or computer is used to perform calculations = round off error.

Representing in binary - 64 bit:
- 1^{st} bit → sign indicator. (0 = +ve, 1 = -ve)
- 11-bit exponent, c → characteristic (range = 0 to 2047)
- 52-bit binary fraction, f → mantissa

base = 2

→ for ensuring equal representability of small nos 0-2047 shifted to -1023 to +1024.

→ f = floating point no. of the form :-
$$(-1)^s \, 2^{c-1023} (1+f)$$

52 binary digits → 16-17 decimal digits.
∴ atleast 16 digit precision.

→ Smallest & Largest Representable numbers:-
↳ $s = 0, c = 1, f = 0$ → $s = 0, c = 2046, f = 2^{-52}$

⇒ $2^{-1022} \times (1+0)$ ⇒ $2^{1023} \times (2 - 2^{-52})$

$\approx 0.2251 \times 10^{-307}$

nos higher than this → overflow.
↓
Computation stops

nos. smaller → underflow

Floating point representation:-

$$0.d_1 d_2 \ldots d_k \times 10^n$$

all $d_i \in \{0, 1, \ldots, 9\}$ except $1 \le d_1 \le 9$

for $y = 0.d_1 d_2 \ldots d_k d_{k+1} \ldots \times 10^n$

Chopping $\rightarrow$ $fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n$

Rounding $\rightarrow$ $fl(y) = 0.\delta_1 \delta_2 \ldots \delta_k \times 10^n$; by adding $5 \times 10^{n-(k+1)}$ to y & chop the result.

for π :- 5 digit

chop: 0.31415×10^1

round: 0.31416×10^1

approx $\rightarrow$ errors.

let p = real no. , p^* = approx.

absolute error $\rightarrow$ $|p - p^*|$ (can be misleading)

relative error $\rightarrow$ $\dfrac{|p - p^*|}{p}$ (more meaningful)

$\Rightarrow$ we say p^* approximates p to t significant digits if t = largest non-negative integer for which

$$\frac{|p - p^*|}{p} < 5 \times 10^{-t}$$

'lub's of absolute error when p^* approximates p to 4 significant digits

$p \rightarrow$	0.1	0.5	100	1000	9990	10000
$\max \lvert p - p^* \rvert \rightarrow$	0.000005	0.00025	0.05	0.5	4.995	5

Arithmetic in Machines

$x \oplus y := fl(fl(x) + fl(y))$

$x \ominus y := fl(fl(x) - fl(y))$

$x \otimes y := fl(fl(x) \times fl(y))$

$x \oslash y := fl(fl(x) \div fl(y))$

Lecture 3.

Sources of Errors in machine calculation:

1. Cancellation of nearly Equal numbers

Ex: $x = \frac{1}{3}$ & 0.3333×10^0

in 5 digit Arithmetic.

$fl(x) = 0.33333 \times 10^0$; $fl(y) = 0.33330 \times 10^0$

$\therefore x \ominus y = 0.0003 \times 10^0 = 0.3 \times 10^{-4}$

Absolute error $= \frac{0.00001}{3}$ and relative error $= \underbrace{\frac{1}{10}}_{\text{reasonably large}}$

2. Division by small numbers / Multiply by large number

let a = real no. $fl(a) = a + \underset{\text{Error}}{\underset{\downarrow}{\varepsilon}}$

$a \div b$, $b = 10^{-n}$

in Machine: $a \oslash b = fl\left(\frac{fl(a)}{fl(b)}\right) = (a+\varepsilon) \times 10^n$

absolute error in calculation $= |\varepsilon| \times 10^n$.

3. in quadratic roots $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

when $b^2 \gg 4ac$.

to get around, rationalize.

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \times \frac{(-b - \sqrt{b^2 - 4ac})}{(-b - \sqrt{b^2 - 4ac})}$$

$$= \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

and other root:-

$$\frac{-2c}{b - \sqrt{b^2 - 4ac}}$$

'b's won't cancel each other here, anyhow, if $b \ggg 4ac$,
Alternate formula results in huge error.
& sign of b determines formula to be used.

→ Errors Propagate easily

How to avoid?
- depends on case at hand, but they vary as cases vary.

Ex:

$ax^3 + bx^2 + cx + d$ → high error
$x(x(ax+b)+c)+d$ → low error. (Nested Polynomial).

Lecture 4
Equations in one Variable

Roots of eqn's: x s.t $f(x) = 0$.
→ Equivalent to solving for $f(x) = \alpha$.

1. Bisection Method:
IVT: Intermediate value theorem: if $f: [a,b] \to R$ (cts) with $f(a) \cdot f(b) < 0$,
then there's a $c \in [a,b]$ s.t $f(c) = 0$.

↑

Method uses this to find sequence whose limit = root of f.
↳ calls for repeated halving subintervals of $[a,b]$

$a_1 = a$, $b_1 = b$ ⇒ $p_1 = \frac{a_1 + b_1}{2}$

Check if p_1 = root, else check sign and perform binary search again, accordingly.

Tip: Keep initial interval smaller.

Termination:

$\{P_i\}$ will converge to root of f.
but none will be a root (for instance).
$\because$ all will be rational nos.

3 criteria to stop.

- $|f(p_n)| < \varepsilon$ (Not good, long time to achieve)
- $|p_n - p_{n-1}| < \varepsilon$ (Not good, doesn't take f into account)
- $p_n \neq 0$ & $\frac{|p_n - p_{n-1}|}{|p_n|}$ (Enticing, but not take f into account)

often used criterion = 2nd one $\Rightarrow |p_n - p_{n-1}| < \varepsilon$.

drawbacks of Bisection Method :-
- Relatively slow to converge.
- Good intermediate approximation inadvertently discarded.

But it :-
- always converges to a solⁿ
- $\therefore$ used as starter for more efficient methods.

Lecture 5

fixed points & roots :- fixed point p, s.t $f(p) = p$

p is a root of $f(x) = 0$ if p is fixed point of $g(x) = x + f(x)$

Fixed Point theorem:
- if $f: [a,b] \to [a,b]$ is continuous, f has a fixed point
- in addition, if $f'(x)$ exists on (a,b) and $|f'(x)| \leq k < 1$ $\forall\ x \in (a,b)$, then f has a unique fixed point in $[a,b]$

→ No proof in this course.

Page No.
Date:

Fixed point iteration: (FPI)

Take any initial approximation $p_0 \in [a,b]$ & generate $\{p_n\}$ s.t $p_i = f(p_{i-1}) \Rightarrow \{p_1, f(p_1), f^2(p_1), f^3(p_1), \dots\}$

$\{p_n\}$ is bounded, but not sure whether convergent.

if $\{p_n\}$ converges,

$$f(p) = f\left(\lim_{n\to\infty} p_n\right) = \lim_{n\to\infty} f(p_n) \quad \Big\} \text{due to continuity}$$

$$= \lim_{n\to\infty} p_{n+1} = p$$

$\Rightarrow$ p is fixed point of f.

Q: how to find fixed point problem that produces a sequence that reliably & rapidly converges to a sol^n to a given root-finding problem?

A: Manipulate question to FPP that satisfies conditions of FPT & has derivative as small as possible near FP.

Lecture 6

Newton-Raphson method: particular FPI method

let $f : [a,b] \to \mathbb{R}$, $f'(x)$ & $f''(x)$ exist.

let $p \in [a,b]$ be a sol^n of $f(x) = 0$

if p_0 is another point in $[a,b]$, then taylor's theorem gives

$$0 = f(p) = f(p_0) + (p-p_0)f'(p_0) + \frac{(p-p_0)^2}{2} f''(\xi)$$

for som ξ between p & p_0

we assume that $|p - p_0|$ is very small,

so $(p-p_0)^2 \approx 0$

$\Rightarrow \quad 0 \approx f(p_0) + (p-p_0)f'(p_0)$

$$\Rightarrow \quad \boxed{p \approx p_0 - \frac{f(p_0)}{f'(p_0)}}$$

initial approximation p_0 & generates sequence $\{p_n\}$ by

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

Note: NR iteration method stops if we get $f'(p_n) = 0$ for some n.

Visualisation :- Successive tangents.

→ Newton Rhapson method is better than fixed point iteration method.

Note $|p-p_0|$ needs to be small so $(p-p_0)^2$ can be dropped from taylor's polynomial.

Theorem: Let $f: [a,b] \to \mathbb{R}$ be twice differentiable.
if $p \in (a,b)$ is such that $f(p) = 0$ & $f'(p) \neq 0$, then there exists a $\delta > 0$ s.t $\forall$ $p_0 \in [p-\delta, p+\delta]$ NR method generates a sequence $\{p_n\}$ converging to p.

Lecture 7.

Problems with Newton-Raphson Method.

- → have to compute f' at each step.
- → f' computation is far more difficult than f - computation.

slight variation :-

by defn: $f'(a) = \lim_{x \to 0} \frac{f(a) - f(x)}{a - x}$

assume p_{n-2} is close to p_{n-1}, then

$$f'(p_{n-1}) \approx \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}$$

using it in NR :-

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$$

→ Called "Secant Method"

Geometrical interpretation :- 2 initial approximations p_0 & p_1

& p_2 = x-intercept of line joining $(p_0, f(p_0))$ & $(p_1, f(p_1))$.

p_3 = — " — " — $(p_1, f(p_1))$ & $(p_2, f(p_2))$.

Only function evaluation is needed per step, not derivative.

→ Method of false position :-

NR & Secant method give successive approximations on one side of the root, i.e $f(p_{n-1}) \cdot f(p_n)$ needn't be -ve then, root might not lie b/n p_{n-1} & p_n.

→ ∴ take approximations which are on both sides of the root.

→ Regula-falsi method :-

Choose p_0 & p_1 s.t $f(p_0) \cdot f(p_1) < 0$.

p_2 chosen in same manner as in Secant method, then compute $f(p_2) \cdot f(p_1)$. if $f(p_2) \cdot f(p_1) < 0$, apply secant method to (p_1, p_2).

if not,
$f(p_2) \cdot f(p_0)$ must be < 0 & hence we apply Secant method to the pair (p_0, p_2).

⇒ root is always bracketed by successive approximations.

Lecture 8

normal practice:- ① use bisection method first
② then use NR or Secant method.

Order of Convergence: to compare rate of convergences of different algorithms.

let $\{p_n\}$ be a sequence converging to p with $p_n \neq p \ \forall n$
if there are +ve constants λ & α such that

$$\lim_{n \to \infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda$$

then, order of convergence of $\{p_n\}$ to p is α with asymptotic error λ.

iterative technique $p_n = g(p_{n-1})$ is said to be of order α if $\{p_n\}$ converges to sol^n $p = g(p)$ with order α.

Order α pace of convergence
λ affects speed of convergence but not to the extent of order (α).

2 cases of order that are significant:

- $\alpha = 1$ & $(\lambda < 1)$ → linearly convergent
- $\alpha = 2$, → quadratically convergent.

$\alpha = 2$ is much better than $\alpha = 1$.

Convergence of fixed point iteration method → linear if $f'(p) \neq 0$.
with $\lambda = |f'(p)| \neq 0$

$\therefore$ for higher order of convergence, $f'(p)$ should be 0.

Theorem: let p ∋ sol^n of $x = f(x)$
let $f'(p) = 0$ & f'' is cts with $|f''(x)| < M$ nearby p.
then there exists $\delta > 0$ s.t for $p_0 \in [p-\delta, p+\delta]$, the sequence defined $p_n = f(p_{n-1})$ converges atleast quadratically to p.

Moreover, for sufficiently large values of n,
$$|p_{n+1} - p| < \frac{M}{2} |p_n - p|^2$$

$\therefore$ for higher order of convergence,
have $f(p) = p$ & $f'(p) = 0$.
for root finding prob, $g(x) = 0$,
construct
$$f(x) = x - \underbrace{\phi(x)}_{\text{chosen later}} g(x)$$

$$f'(x) = 1 - \phi'(x) g(x) - \phi(x) g'(x)$$
$$0 = f'(p) = 1 - \phi(p) g'(p) \Rightarrow \phi(p) = \frac{1}{g'(p)}$$

$$\Rightarrow p_{n+1} = f(p_n) = p_n - \frac{g(p_n)}{\underbrace{g'(p_n)}_{\neq 0}} \Rightarrow \text{NR method.}$$

M T W T F S S
Page No.:
Date:
YOUVA

Multiplicity of a zero.

$g: [a,b] \to \mathbb{R}$ is fn & let $p \in [a,b]$ be zero of g

p = zero of multiplicity m if

$$g(x) = (x-p)^m q(x) \text{ with } \lim_{x \to p} q(x) \neq 0$$

Simple zero → $m=1$, NR works.
doesn't work for $m>1$

Lecture - 9

Order of Fixed Point Iteration method:
root finding problem $g(x)=0$ with $x=p$
converted to FPP with $f(x)$.

if $f'(p) \neq 0$ → Order = linear
if $f'(p) = 0$ → Order = Quadratic / higher.

if $x=p$ is simple zero of g → NR → Quadratic,
if not, No guarantee of Quadratic convergence.

Modified NR method:

$g: [a,b] \to \mathbb{R}$, $x=p$ → root of f

def: $\mu(x) = \dfrac{g(x)}{g'(x)}$

Order of $x=p$ as g's zero $= m \Rightarrow g(x) = (x-p)^m q(x)$

$g'(x) = m(x-p)^{m-1} q(x) + (x-p)^m q'(x)$

$$\mu(x) = x-p \left[\frac{q(x)}{mq(x) + (x-p)q'(x)} \right]$$

p = Simple zero of $\mu(x)$.
further, if $g(x)$ has no zero in neighborhood of $x=p$, $\mu(x)$ also has no zero

$\therefore$ Apply NR to $u(x)$

$$p_{n+1} = f(p_n) \text{ where } f(x) = x - \frac{u(x)}{u'(x)}$$

$$= x - \frac{g(x)g'(x)}{(g'(x)^2 - g(x)g''(x))}$$

Pro :- → converges to p atleast quadratically.

→ drawback :- • computation of $g''(x)$ also.

• Cancelling of nearly equal terms in denominator.

Other methods to improve order of convergence :-

1. Aitken's Δ^2-method to accelerate convergence :-

if $\{p_n\}$ converges to p,

for large enough n, $(p_{n+1} - p)^2 \approx (p_n - p)(p_{n+2} - p)$

$$\Rightarrow p \approx p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} = \hat{p}_n$$

Lecture 10

Interpolation.

→ finding f that fits given data & find values of all points inbetween.

use polynomials :- $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

↳ approximate continuous functions uniformly.

Note : Given any $f : [a,b] \to \mathbb{R}$, there exists a polynomial that is as "close" to given function as desired.

↳ Weierstrass approximation theorem.

with P, computations are easier as derivatives are polynomials too.

→ Natural choice : Taylor, but they approximate only at a point.

Misconception : higher degree Taylor polynomials approximate function better ✗ NO! not true.

$ex: f(x) = 1/x$.

reason for failure :- only value of f & derivatives at x_0 is considered.

→ use method which includes info about various other point.

→ Lagrange interpolating polynomial :-

for $f(x_0) = y_0$, simplest approx P is $P(x) = y_0$

for $f(x_0) = y_0$

$f(x_1) = y_1$

linear polynomial $P(x) = a_1 x + a_0$

$a_1 x_0 + a_0 = y_0$

$a_1 x_1 + a_0 = y_1$

$$\Rightarrow \quad a_1 = \frac{y_0 - y_1}{x_0 - x_1} \; ; \; a_0 = \frac{y_1 x_0 - y_0 x_1}{x_0 - x_1}$$

$$\Rightarrow \quad P(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) y_0 + \left(\frac{x - x_0}{x_1 - x_0}\right) y_1$$

require $x_0 \neq x_1$

∴ To generate polynomials :-

PTO

Let $x_0, x_1, \dots, x_n$ be distinct $(n+1)$ points

&

$f(x_0) = y_0,\ f(x_1) = y_1, \dots, f(x_n) = y_n$

find P with $P(x_i) = y_i \quad \forall\ i = 0, \dots, n$.

Solve $y_i = \delta_{i,j}$ find $L_{n,i}$ with

$$L_{n,i}(x_j) = \delta_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

for fixed i, $L_{n,i}(x_j) = 0 \quad \forall\ j \neq i \Rightarrow (x - x_j)$ divides $L_{n,i}(x)$

$\because x_j$ are distinct,

$(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)$ divides $L_{n,i}(x)$

no x_i

$\therefore$ define

$$L_{n,i}(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

$$\Rightarrow L_{n,i}(x_j) = \delta_{i,j}$$

$$\therefore P(x) = y_0 L_{n,0}(x) + y_1 L_{n,1}(x) + \dots + y_n L_{n,n}(x)$$

Lecture 11

→ degree of Lagrange interpolation polynomial is atmost no. of points − 1

Uniqueness of interpolating polynomial? :-

The no. of polynomials we can get to interpolate given function at $(n+1)$ points $= \infty$

But, if degree is restricted to be $\leq n$, we have unique polynomial.

Theorem :- polynomial of degree n has atmost n distinct zeros.

$\Rightarrow$ polynomial of degree $\leq n$ with $(n+1)$ zeroes = zero polynomial

Error by interpolating polynomial:-

$f:[a,b] \to \mathbb{R}$ be $(n+1)$ times ctsly differentiable. let $P(x)$ be polynomial interpolating f at n+1 distinct pts

$x_0, x_1, \ldots, x_n \in [a,b]$

then, $\forall\ x \in [a,b]$, $\exists\ \xi(x) \in (a,b)$ with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

Lecture 12

Practical difficulty:- for calculating step size to ensure max error, we don't need explicit linear polynomial interpolating the function, but we use full info about f.

BUT, we only will have datapoints & no info about f.

→ We ~~as~~ also need a method to ~~cal~~ cumulatively calculate the interpolating polynomials.

For this,

Constant polynomial for node x_0 will be $P_0(x) = f(x_0)$ &

x_1 ——— $Q_0(x) = f(x_1)$

∴ Linear polynomial :-

$$P_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

$$= \frac{(x - x_1) P_0(x) - (x - x_0) Q_0(x)}{x_0 - x_1}$$

Now, consider Quadratic case also :-

$$Q_1(x) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2)$$

$$P_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1)$$

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

Manipulation Gives

$$P_2(x) = \left(\frac{x - x_2}{x_0 - x_2}\right)\left[\frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)\right]$$

$$- \left(\frac{x - x_0}{x_0 - x_2}\right)\left[\frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2)\right]$$

$$\boxed{P_2(x) = \frac{(x - x_0) P_1(x) - (x - x_0) Q_1(x)}{x_0 - x_2}}$$

If P_2 interpolates f on $\{x_0, x_1, x_2\}$
Q_2 on $\{x_1, x_2, x_3\}$ & P_3 on $\{x_0, x_1, x_2, x_3\}$
then do we get

$$P_3(x) = \frac{(x-x_3)P_2(x) - (x-x_0)Q_2(x)}{x_0 - x_3} \quad ?$$

→ Yes.

Lecture 13

Neville's formula

Let f be defined on $\{x_0, x_1, \dots, x_n\}$.
choose 2 distinct nodes x_i & x_j

Let

Q_i : Polynomial interpolating f on all nodes except x_i
Q_j : — " — — " — x_j

the P : polynomial interpolating f on all nodes,

$$P(x) = \frac{(x-x_j)Q_j(x) - (x-x_i)Q_i(x)}{x_i - x_j}$$

Table :

x_0	P_0				
x_1	P_1	$P_{0,1}$			
x_2	P_2	$P_{1,2}$	$P_{0,1,2}$		
x_3	P_3	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$	
x_4	P_4	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$

Divided differences:

Another interpolation method.

Given f on distinct $(n+1)$ nodes $x_0, \dots, x_n$ there's unique polynomial P_n interpolating f on these nodes.

define $f[x_0, \dots, x_n]$ as coeff of x^n in P_n.

$(n+1)$th order divided difference.

doesn't depend on ordering of nodes.

$\therefore$ Get recurrence formula for coeff $f[x_0, \dots, x_n]$.

P_{n-1} : interpolates on $x_0, x_1, \dots, x_{n-1}$

Q_{n-1} : — " — $x_1, x_2, \dots, x_n$

$$\Rightarrow \quad P_n(x) = \frac{(x-x_0)Q_{n-1}(x) - (x-x_n)P_{n-1}(x)}{x_n - x_0}$$

$$\text{Coeff of } x^n \text{ in } P_n = \frac{(\text{Coeff of } x^{n-1} \text{ in } Q_{n-1}) - (\text{Coeff } x^{n-1} \text{ in } P_{n-1})}{x_n - x_0}$$

$$f[x_0, \dots, x_n] = \frac{f[x_1, x_2, \dots x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}$$

$\rightarrow$ Get recurrence relⁿ for P_n in terms of divided differences.

Note $\forall\, i<n$ $P_n(x_i) = P_{n-1}(x_i)$

$\Rightarrow$ $P_n - P_{n-1}$ has zero at $x_0, x_1, \dots, x_{n-1}$

$\therefore$

$$P_n - P_{n-1} = \alpha(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

real number.

has to be coeff of monomial x^n in P_n as $P_{n-1} \leq n-1$

$\Rightarrow f[x_0, x_1, \dots, x_n] = \alpha$

$$P_n = P_{n-1} + (x-x_0)(x-x_1)\dots(x-x_{n-1})\, f[x_0, x_1, \dots, x_n]$$

Computing the divided differences:-

$$\begin{array}{llll}
x_0 & f(x_0) & & \\
 & & f[x_0,x_1] & \\
x_1 & f(x_1) & & f[x_0,x_1,x_2] \\
\vdots & & f[x_1,x_2] & \quad \vdots \qquad\qquad f[x_0,\dots,x_n] \\
x_2 & f(x_2) & & f[x_{n-2},x_{n-1},x_n] \\
\vdots & & f[x_{n-1},x_n] & \\
x_n & f(x_n) & &
\end{array}$$

Can go in any-way along the tree, two such ways:-

1) Forward formula:

$$P_n(x) = f(x_0) + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + \dots + f[x_0,x_1,\dots,x_n](x-x_0)\cdots(x-x_{n-1})$$

2) Backward formula:-

$$P_n(x) = f(x_n) + f[x_{n-1},x_n](x-x_n) + f[x_{n-2},x_{n-1},x_n](x-x_n)(x-x_{n-1}) + \dots + f[x_0,x_1,\dots,x_n](x-x_n)\dots(x-x_1)$$

$\rightarrow$ Forward formula in Nested form:

$$P_n(x) = f(x_0) + (x-x_0)\big[f[x_0,x_1] + (x-x_1)\big[f[x_0,x_1,x_2] + (x-x_2)\dots$$

ex:

$$P_2(x) = f(x_0) + (x-x_0)\Big[f[x_0,x_1] + (x-x_1)f[x_0,x_1,x_2]\Big]$$

in def^n of $f[x_0,\dots,x_n]$ we need all x_i to be distinct.
but when some nodes are equal to each other,

General Mean value theorem:

f is n-times continuously differentiable; on $[a,b]$

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

for some $\xi \in [a,b]$

→ define $f[x_0, x_0] = f'(x_0) = \lim_{x_1 \to x_0} f[x_0, x_1]$

∴ define

$$f[x_0, x_1, x_0] = f[x_0, x_0, x_1] = \frac{f[x_0, x_1] - f[x_0, x_0]}{x_1 - x_0}$$

$$= \frac{f[x_0, x_1] - f'(x_0)}{x_1 - x_0}$$

using general MVT :-

$$f[x_0, x_0, x_0] = \frac{f^{(2)}(x_0)}{2}.$$

∴ $f[x_0, \dots, x_n]$ is defined in general.

Now, by letting x_n as x, we get divided differences as a function of x :-

$$f[x_0, x_1, \dots, x_{n-1}, x]$$

this function is continuous. Indeed,

$$f[x_0, x] = \begin{cases} \dfrac{f(x) - f(x_0)}{x - x_0} & x \neq x_0 \\ f'(x_0) & x = x_0 \end{cases}$$

$\Rightarrow f[x_0, x]$ is cts.