CSE 330

Numerical Methods

[AQV]

Floating Point Arithmetic - 1

18/02/2025
Tuesday

Direct Method

$x + 2y = 0$ ---(i)
$2x - ny = 1$ ---(ii)
$2x + 4y = 0$ (x)
$-2x + ny = -1$
(y)

Iterative Method

$\sin(1.2) = 1.2 - \frac{(1.2)^3}{3!} + \frac{(1.2)^5}{5!} - \frac{(1.2)^7}{7!} + \cdots$
(infinite terms)
$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$
(using taylor series)

1	2	3	4	Term

1st term 1.2
2nd term → 0.912
3rd term → 0.93276

→ Truncation Error

Due to computer limitations, the value is approximated. The parts which is left out, is called Truncation Error

Eliminating terms from an infinite no. of terms

Back in Babylonian civilization history

*** Different computers have different capabilities

eg. Computer 1: $\pi = 3.14159$
Computer 2: $\pi = 3.14$
} Floating Point Approximation

Eliminating values after certain decimal points

eg. 2.5762 (actual value), 2.58 (approximated by a computer)

∴ 2.58
−2.5762
0.0038 → Rounding Error

Floating point arithmetic

Fixed point

$(10.215)_{10}$

$x = \pm (d_1 d_2 d_3 \cdots d_{k-1} \cdot d_k \cdots d_n)\beta$
(· = point; fraction; β = base)

Floating point

10.245
0.10215×10^{2} (2 = exponent)

eg. $(10.1)_2 = 1\times2^1 + 0\times2^0 + 1\times2^{-1}$
$\beta = (2.5)_{10}$
β^e

Floating point Numbers

FCR $(-\infty, +\infty)$

$f = \pm (0 \cdot d_1 d_2 \cdots d_m)\beta^e$ ($d_1 \neq 0$)

± → sign; $(0 \cdot d_1 d_2 \cdots d_m)$ → fraction or mantissa or significand; β → base; e → exponent

$e_{min} \le e \le e_{max}$ due to computer limitation

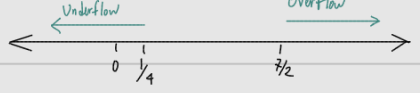
no. of digits after decimal

$\beta = 2$, $m = 3$, $e_{min} = -1$, $e_{max} = 2$
$e = \{-1, 0, 1, 2\}$

$0 \cdot \square\square\square \times 2$ (0/1 0/1 0/1)

$(0.111)_2 \times 2^2 = 7/2$ (highest possible value)

$(0.100)_2 \times 2^{-1} = 1/4$ (lowest possible value)



Total no. of values:

$0.100 \times 2^{-1} \Rightarrow 1/4$
$0.101 \times 2^{-1} \Rightarrow 5/16$
$0.110 \times 2^{-1} \Rightarrow 3/8$
$0.111 \times 2^{-1} \Rightarrow 7/16$
} e = −1, 4 values

For e = −1, 0, 1, 2 → Total 16
16 for +ve value
16 for −ve value
∴ Total 32 values

Floating Point Arithmetic - 2

IEEE Standard: To represent floating point number we'll use 64 bit architecture

Sign	Exponent	fraction
1 bit	11 bit	52 bits

$(0 \cdot d_1 d_2 \cdots d_m)_\beta \times \beta^e$ (52 bits)

$2^{11} = 2048$

$e_{min} = 0$
$e_{max} = 2047$
} only +ve
Total 2048 (including 0)

$e_{min} = -1023$
$e_{max} = 1024$
} +ve & −ve value
Total 2048 (including 0)

$F = \{ \pm (0 \cdot d_1 d_2 \cdots d_m)_\beta \times \beta^e \mid \beta, d_i, e \in \mathbb{Z},\ 0 \le d_i \le \beta - 1,\ e_{min} \le e \le e_{max} \}$ (e = exponent)

$2^3 = 8$ ∴ 0 to 7
for if $\beta = 10$ $0 \le d_i \le 9$

Convention 1: Standard Form: $F = \pm (0 \cdot d_1 d_2 \cdots d_m)_\beta \times \beta^e$ where $d_1 \neq 0$ (m)

Convention 2: Normalized: IEEE $F = \pm (0.1 d_1 d_2 \cdots d_m)_\beta \times \beta^e$ (m)
fraction / significand / mantissa (m)

In prev semesters normalized & denormalized used to have the name of the other

Convention 3: Denormalized IEEE $F = \pm (1 . d_1 d_2 \cdots d_m)_\beta \times \beta^e$ (m)

Fixed-point numbers

$x = \pm (d_1 d_2 d_3 \ldots\ldots d_{k-1} . d_k \ldots\ldots d_n)_\beta$

where,
d = digit, β = number systems base, $d_1 \ldots\ldots d_n \in \{0, 1, 2 \ldots\ldots \beta - 1\}$

Example:
$(10.1)_2 \Rightarrow (d_1 d_2 . d_3)_\beta = 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} = (2.5)_{10}$

$(-3.14)_2 \Rightarrow -(d_1 . d_2 d_3)_\beta = -(3 \times 2^0 + 1 \times 2^{-1} + 4 \times 2^{-2}) = (-4.5)_{10}$

Floating-point numbers

$F = \{ \pm (0 . d_1 d_2 \ldots\ldots d_m)_\beta \times \beta^e \mid \beta, d_i, e \in Z, 0 \le d_i \le \beta - 1, e_{min} \le e \le e_{max} \}$

Here, $(0 . d_1 d_2 \ldots\ldots d_m)_\beta$ is called the fraction or significand or mantissa,
β is the base, and e is the exponent.

Example:
$(123.45)_2 = 0.12345 \times 10^3$

$(11.101)_2 = 0.11101 \times 2^2$

Cost of using Floating Point number

1) Numbers are not equally spaced
2) We can only represent a finite set of numbers

IEEE standard (1985) for double-precision (64-bit) arithmetic

Here β = 2, and there are 52 bits for the fraction, 11 for the exponent, and 1 for the sign. The actual format used is-

(Denormalized Form)

here, the e (exponent) range is [0, 2047] since

The largest possible number is

The smallest possible number is (not small; it cannot express 0.001)

If we want numbers less than 1, we can do **exponent bias**

Example:

Now, the e (exponent) range is [-1022, 1025]

The largest possible number is

The smallest possible number is

Actually, the exponents −1022 and 1025 are used to store , respectively.

So,

The largest possible number (except) is

The smallest possible number (except) is

Significant figures

The concept of significant figures (s.f.) helps us express numbers with the right level of precision.

- How to Count Significant Figures?
  - ✓Start counting from the first non-zero digit.
  - ✓Count all digits after that, including zeros if they come after the decimal point.

Example:

3.1056 → 5 s.f. (all digits count)

31.050 → 5 s.f. (the final zero counts because it's after the decimal)

0.031056 → 5 s.f. (leading zeros don't count, but everything after '3' does)

0.031050 → 5 s.f. (final zero still counts)

3105.0 → 5 s.f. (the zero counts because of the decimal)

- How to Round to Significant Figures?

  To round a number to n significant figures, find the first n digits, then round off.

Example:

3.14159 rounded to 3 s.f. → 3.14

0.007823 rounded to 2 s.f. → 0.0078

45678 rounded to 3 s.f. → 45,700 (notice the zeros replace rounded-off digits)

However, significant figures help keep calculations accurate without unnecessary detail. For example, if a measurement is 2.36 cm, writing 2.3600 cm suggests extra precision that wasn't actually measured.

Rounding

If number is 5, the number is rounded to nearest even number.

11·16 → 11·2

11·14 → 11·1

11·15 → 11·2

what we actually do

11·15 → 11·2 (even)

11·1

11·25 → 11·3 (even)

11·2

What computer does (due to IEEE standards)

Rounding

For binary, if the number ends in 0 → even and if the number ends in 1 → odd

Example: Round and numbers where [Convention 01]

Example: Round numbers where [Convention 01]

Example: Round numbers where [Convention 01]

Significant figures (S.F)

3.10567 → 3.1057 → 5 s.f

031.0567 (6 s.f) → 31.057 (5 s.f)

0.0310567 (6 s.f) → 0.03|1057 = 0.31057 × 10^-1 (0.d1d2...)

Rounding Error

x = 21.531 ⇒

fl(x) = 21.53 ⇒

Error 0.001

Error = |fl(x) − x|

Relative error (δ) = |fl(x) − x| / |x| (for percentage error) × 100

Scale Invariant error

for maximum scale invariant error, δmax = |fl(x) − x| / |x|max

eg x = 21.535 (computer standard)

fl(x) = 21.54

0.005 ⇒ 0.5 × 10^-2

(0.d1d2d3...)β × β^e ⇒ ½ × β^-m

eg x = 21.55

fl(x) = 21.6

0.05 ⇒ 0.5 × 10^-1

eg 21.53 → 21.5 → 0.03

eg 21.54 → 21.5 → 0.04

eg 21.57 → 21.60 → 0.03

Max^m Error = ½ β^-m

Max^m Possible Total Error = ½ × β^-m × β^e

eg 0.2153 5 × 10^2 → x

0.2154 0 × 10^2 → fl(x)

error → 0.0000 5 × 10^2 = 0.5 × 10^-4 × 10^2

½ × β^-m × β^e

|fl(x) − x| ≤ ½ β^-m β^e

|fl(x) − x| / |x| ≤ ½ β^-m β^e / |x|

δ ≤ ½ β^-m β^e / |x| ← lowest

For δ to be max^m possible value

(0.d1d2d3...)β × β^e = 0.1000... × β^e

x = 1/10 β^e

= 10^-1 β^e = β^-1 × β^e

δ ≤ ½ β^-m β^e / β^-1 β^e

∴ δ ≤ ½ β^1-m → Max^m Relative error

Em (Epsilon m) [Machine Epsilon]

Machine Epsilon or Unit roundoff: Maximum scale invariant rounding error (which varies according to convention)

For [Standard Form or Convention 1],

Machine Epsilon or Unit roundoff,

For [Normalized Form or Convention 2],

Machine Epsilon or Unit roundoff,

For [Denormalized Form or Convention 3],

Machine Epsilon or Unit roundoff,

Loss of significance

Using x = (−b ± √(b²−4ac)) / 2a

x² − 56x + 1 = 0

x1 = 28 + √783 = 55.98

55,98213711

55,98

0.00203216

x2 = 28 − √783 = 28 − 27.98 = 0.02

28

27.98213 7

0.01786 actually

27.98 2137 actually

0.02

0.01786

0.01786

Using Middle term

x² − (x1 + x2)x + (x1x2) = 0

x1x2 = 1 ⇒ x2 = 1/x1

This way more accurate

x1 = 55.98 x2 = 1/55.98 = 0.01786

When we subtract two close numbers, loss of significance increases

But when we add, loss of significance is much less

Polynomial Interpolation-1

21/02/2025
Friday

This chapter deals with functions.
The computer recognizes a function by converting it to a polynomial.
We'll learn how this is done.
We recognize function as e.g. $f(x) = \sin x$ and computer as e.g. $P_n(x)$

Polynomial: Series of terms. e.g. $2x^2 - 5x + 3$
→ n means degree (highest power of the series of terms)
→ Exponent must be Non-negative & an Integer.

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_n x^n$$
where $\{a_0, a_1, a_2, \dots a_n\}$ are coefficients/constants
$x \Rightarrow$ Variable
$n \Rightarrow$ degree

Number of Terms
eg $7x^7 - 2x^3$
max no. of terms with degree 7: $7x^7 + 0.x^6 + 0x^5 + 0x^4 + 0x^3 + 0x^2 + 0x + 0.x^0$ [8 terms]
∴ for n degree
→ (n+1) terms max
→ (n+1) coefficients max

eg $f(x) = \sin x$
$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ (Truncation error)
$P_6(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + 0.x^6 -$
This computer knows till degree 6

degree ↑ Accuracy ↑

As computer has limitations,
$f(x) \neq P_n(x)$
eg $\sin x \neq x - \frac{x^3}{3!} + \frac{x^5}{5!}$
∴ $|f(x) - P_n(x)|$ = Truncation Error

Taylor series

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0)^1 + \frac{f''(x_0)}{2!}(x-x_0)^2 \dots$$
$$= \sum_{k=0}^{n} \frac{f^k(x_0)}{k!}(x-x_0)^k + \frac{f^{n+1}(x_0)}{(n+1)!}(x-x_0)^{n+1} + \frac{f^{n+2}(x_0)}{(n+2)!}(x-x_0)^{n+2} + \dots$$
If there was no error; n = ∞
Error starts ↓
Error Bound = $\frac{f^{n+1}(\xi)}{(n+1)!}(x-x_0)$; ξ_i

eg $f(x) = \sin x$, $n = 6$
$P_6(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!}$
∴ Error Bound = $\frac{f^{6+1}(\xi)}{(6+1)!}(x-x_0)^{6+1}$
$= \frac{-\cos(\xi)}{7!}(x-x_0)^7$ //

$f(x) = \sin x$
$f'(x) = \cos x$
$f''(x) = -\sin x$
$f'''(x) = -\cos x$
$f^{IV}(x) = \sin x$
$f^{V}(x) = \cos x$
$f^{VI}(x) = -\sin x$
$f^{VII}(x) = -\cos x$

Points/nodes ↑ Accuracy ↑

Interpolation is a mathematical technique used to estimate unknown values within a known set of data points. It involves constructing a function that passes through or closely follows given data points to predict intermediate values.

Types of Interpolation:

1. **Linear Interpolation** – Estimates values using a straight line between two known points.
2. **Polynomial Interpolation** – Uses polynomials (e.g., Lagrange or Newton's interpolation) to fit the data.
3. **Spline Interpolation** – Fits piecewise polynomials (e.g., cubic splines) to ensure smooth transitions between data points.

Applications:

- Image processing (resizing images)
- Data analysis (filling missing data)
- Engineering (curve fitting)
- Computer graphics (animation and rendering)

Given $(x_0, x_1, x_2, x_3 \dots x_n) \Rightarrow$ n+1 nodes $\Rightarrow$ Polynomial will have degree n
and $f(x) = \square$, find $P_n(x)$

∴ $P_n(x) = a_0 + a_1 x + a_2 x_2 + a_3 x_3 + \dots + a_n x^n$ [Put n+1 points to get n+1 eqⁿ]

$$P_n(x_0) = a_0 + a_1 x_0 + \dots + a_n x_0^n$$
$$\vdots$$
$$P_n(x_n) = a_0 + a_1 x_n + \dots + a_n x_n^n$$
} n+1 eqⁿ

Matrix Representation

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & \vdots & \vdots & & \vdots \\ \vdots & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$
V (Vandermonde matrix) · a (coefficients) = f (function)

$V \cdot a = f$
$\therefore a = V^{-1} f$

Polynomial Interpolation - 2

28/02/2025 Friday

Q) Using Vandermonde matrix find the interpolating polynomial

(i) given $f(x) = 1/x^2$, $x_0 = 2$, $x_1 = 4$ → two nodes

$\# P_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_n x$ (general formula of a polynomial)

#(n+1) nodes → P_n

∴ For two nodes degree is 1 ∴ $P_1(x) = a_0 + a_1 x$

$P_1(x_0) = a_0 + a_1 x_0$

$P_1(x_1) = a_0 + a_1 x_1$

$$\therefore \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \end{pmatrix}$$

V A f

$$\therefore \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \end{pmatrix}^{-1} \begin{pmatrix} f(x_0) \\ f(x_1) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 1/4 \\ 1/16 \end{pmatrix}$$

→ Use Calculator

$$= \begin{pmatrix} 0.4375 \\ -0.09375 \end{pmatrix}$$

$f(x) = \frac{1}{x^2}$

$f(2) = \frac{1}{2^2} = \frac{1}{4}$

$f(4) = \frac{1}{4^2} = \frac{1}{16}$

Only square matrices can have inverse matrix

$\therefore P_1(x) = a_1(x) = a_0 + a_1 x$

$P_1(x) = 0.4375 + (-0.09375)x$ //

(ii) Find interpolation error when $x = 9$

Interpolation Error = $|f(x) - P_1(x)|$

$= |f(9) - P_1(9)| = \frac{1}{9^2} - (-0.40625) = 0.39391$ //

(n+1) nodes → P_n ⇒ Existence / Uniqueness Theorem

→ Till now nodes & P_n were given. Then we use inverse matrix to find the coefficients But, using inverse matrix takes a lot of resources for computer. Takes a lot of processing time. Mathematician Langrange gave a theorem to solve this problem without using inverse matrix.

Vector Space — 2D

$\vec{A} = A_x \hat{i} + A_y \hat{j}$ — $\hat{i}, \hat{j}$ ⇒ basis; coefficients

$P_n = a_0 + a_1 x + a_2 x^2 + \dots a_n x^n$ of polynomial

$\{1, x, x^2, \dots, x^n\}$ → basis

$\{a_0, a_1, a_2, \dots a_n\}$ → coefficient

Langrange Polynomial

given (n+1) nodes, $f(x)$: coefficient; Langrange basis

$$P_n(x) = \sum_{k=0}^{n} f(x_k) \cdot l_k(x)$$

$\{l_0(x), l_1(x), l_2(x) \dots, l_n(x)\}$ → basis

$\{f(x_0), f(x_1), f(x_2) \dots, f(x_n)\}$ → coefficient

eg Given Nodes = 2, $f(x)$ — Σ sum, ∏ product

$n=1$, $P_1(x) = f(x_0) l_0(x) + f(x_1) l_1(x)$

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \left(\frac{x - x_j}{x_k - x_j}\right) = \left(\frac{x - x_0}{x_k - x_0}\right) \cdot \left(\frac{x - x_1}{x_k - x_1}\right)\left(\frac{x - x_2}{x_k - x_2}\right) \cdots \left(\frac{x - x_n}{x_k - x_n}\right)$$

given $f(x) = \cos x$

$x_0 = -\pi/4 \longrightarrow f(-\pi/4) = \frac{1}{\sqrt{2}} = f(x_0)$

$x_1 = 0 \longrightarrow f(0) = 1 = f(x_1)$

$x_2 = \pi/4 \longrightarrow f(\pi/4) = \frac{1}{\sqrt{2}} = f(x_2)$

∴ $P_2(x) = f(x_0) l_0(x) + f(x_1) l_1(x) + f(x_2) l_2(x)$ $n=2$

$$\# l_0(x) = \left(\frac{x - x_0}{x_0 - x_0}\right)\left(\frac{x - x_1}{x_0 - x_1}\right)\left(\frac{x - x_2}{x_0 - x_2}\right) = \left(\frac{x - 0}{-\pi/4 - 0}\right)\left(\frac{x - \pi/4}{-\pi/4 - \pi/4}\right) = \frac{8}{\pi^2} x\left(x - \pi/4\right)$$

(as $j = k$)

$$\# l_1(x) = \left(\frac{x - x_0}{x_1 - x_0}\right)\left(\frac{x - x_1}{x_1 - x_1}\right)\left(\frac{x - x_2}{x_1 - x_2}\right) = \frac{-16}{\pi^2}\left(x + \pi/4\right)\left(x - \pi/4\right)$$

$$\# l_2(x) = \left(\frac{x - x_0}{x_2 - x_0}\right)\left(\frac{x - x_1}{x_2 - x_1}\right)\left(\frac{x - x_2}{x_2 - x_2}\right) = \frac{8}{\pi^2} x\left(x + \pi/4\right)$$

$l_k(x) = \delta$ → knockeri delta when $k = j$

∴ $P_2(x) = f(x_0) l_0(x) + f(x_1) l_1(x) + f(x_2) l_2(x)$ [Put the values and simplify] //

Polynomial Interpolation - 3

\# n+1 nodes (lagrange coefficient, lagrange basis)

$$\# P_n(x) = \sum_{k=0}^{n} f(x_k)\, l_k(x)$$

$$l_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n=1} \frac{x-x_j}{x_k-x_j}$$

eg $f(x) = x^2$, $P_n = ?$

(i) nodes, $x_0 = 2 \rightarrow f(x_0) = 4$

$x_1 = 3 \rightarrow f(x_1) = 9$

as $n = 2$

$\therefore P_1(x) = f(x_0)\, l_0(x) + f(x_1)\, l_1(x)$

$$\# l_0(x) = \left(\frac{x-x_1}{x_0-x_1}\right) = \frac{x-3}{2-3} = -(x-3)$$

$$\# l_1(x) = \left(\frac{x-x_0}{x_1-x_0}\right) = \frac{x-2}{3-2} = (x-2)$$

$\therefore$ Interpolating polynomial $P_1(x) = 4(-(x-3)) + 9(x-2) = -4x+12+9x-18 = 5x-6$ //

(ii) Find the error for $x = 5$

Interpolation error $= |f(x) - P_n(x)| = |(5)^2 - (5(5)-6)| = |25-19| = 6$ //

☆ 8) given the function passes through $(2,4)$ and $(3,9)$ (x_0, $f(x_0)$, x_1, $f(x_1)$)

8) given function passes through $(1,2)$, $(2,4)$, $(3,11)$ how many nodes are there & what's the degree of the polynomial?

= no. of nodes = 3 // degree = 3-1 = 2 //

Lagrange

Advantages	Disadvantages
• Fast Calculation	• Non Dynamic (Recalculation for adding new nodes)

Thus we use a different method → Newton form

Newton / Divided difference form (Dynamic)

node = (n+1)+1 (new) (old)

$P_{n+1}(x) = P_n(x) + g_{n+1}(x)$

eg $\underbrace{x^2+3x+5}_{\text{new}} = \underbrace{x+3}_{\text{old}} + x^2+2x+2$

$$P_n(x) = \sum_{k=0}^{n} f(x_k)\, l_k(x)$$

Newton's Coefficient, Newton's basis

(n+1) nodes : $P_n(x) = \sum_{k=0}^{n} a_k n_k(x)$

$= a_0 n_0(x) + a_1 n_1(x) + a_2 n_2(x) + \cdots a_n n_n(x)$

$= f(x_0)\, n_0(x) + f[x_0,x_1]\, n_1(x) + \ldots + f[x_0, x_1, x_2, \cdots x_n]$

$n_0(x) = 1$

$n_1(x) = (x-x_0)$

$n_2(x) = (x-x_0)(x-x_1)$

$n_3(x) = (x-x_0)(x-x_1)(x-x_2)$

$\vdots$

$n_k(x) = (x-x_0)(x-x_1)\cdots(x-x_{k-1})$

eg given 4 nodes $\{-1, 0, 1, 2\}$

$f(x)$ $\{5, 1, 1, 11\}$

$$P_3(x) = \sum_{k=0}^{3} a_k n_k(x)$$

$= a_0 n_0(x) + a_1 n_1(x) + a_2 n_2(x) + a_3 n_3(x)$

$= f(x_0)\cdot 1 + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2) + f[x_0,x_1,x_2,x_3,x_4](x-x_0)(x-x_1)(x-x_2)(x-x_3)$

when new node added:

$x_0 = -1$ $f[x_0] = 5$

$x_1 = 0$ $f[x_1] = 1$ $\rightarrow f[x_0,x_1] = \frac{f[x_1]-f[x_0]}{x_1-x_0} = -4$ (a_1)

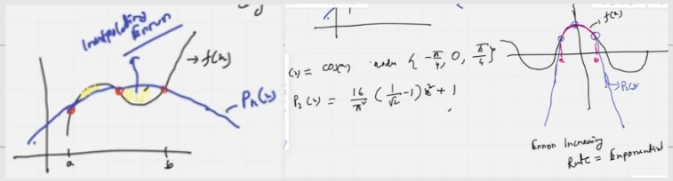
$x_2 = 1$ $f[x_2] = 1$ $\rightarrow f[x_1,x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1} = 0$ $\rightarrow f[x_0,x_1,x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = 2$ (a_2)

$x_3 = 2$ $f[x_3] = 11$ $\rightarrow f[x_2,x_3] = \frac{f[x_3]-f[x_2]}{x_3-x_2} = 10$ $\rightarrow f[x_1,x_2,x_3] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1} = 5$ $\rightarrow f[x_0,x_1,x_2,x_3] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0} = 1$ (a_3)

$x_4 = 5$ $f[x_4] = 15$ $\rightarrow f[x_3,x_4] = \frac{f[x_4]-f[x_3]}{x_4-x_3}$ $\rightarrow f[x_2,x_3,x_4] = \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$ $\rightarrow f[x_1,x_2,x_3,x_4] = \frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$ $\rightarrow f[x_0,x_1,x_2,x_3,x_4]$

Interpolation Error

Polynomial Interpolation - 4



Taylor

$$|f(x) - P_n(x)| = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

Error bound / Lagrange form of the remainder

Cauchy's Theorem

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

eg $f(x) = \cos x$, nodes $= \{-\pi/4, 0, \pi/4\}$, interval $[a,b]$

$$|f(x) - P_2(x)| = \left|\frac{f^3(\xi)}{3!}(x+\pi/4)(x-0)(x-\pi/4)\right|$$

If interval not given, take $a = -\pi/4$, $b = \pi/4$

Max^m possible error / Upper bound of error

$$|f(x) - P_2(x)| \le \frac{f^3(\xi)}{6}\left(x^2 - \frac{\pi^2}{16}\right)x$$

$\to [-1,1]$; $\sin(1) = 0.8415$, $\sin(-1) = -0.8415$ } both same take any as theirs modulus

$$\le \left|\frac{\sin(\xi)}{6}\left(x^3 - \frac{\pi^2}{16}x\right)\right|$$

(i) Max: $\frac{\sin(\xi)}{6}$

(ii) Max: $x^3 - \frac{\pi^2}{16}x$

$\omega(x) = x^3 - \frac{\pi^2}{16}x$

$\omega'(x) = 3x^2 - \frac{\pi^2}{16}$

at max, min point, $\omega'(x) = 0$

$3x^2 - \frac{\pi^2}{16} = 0$

$x = \pm\frac{\pi}{4\sqrt{3}}$

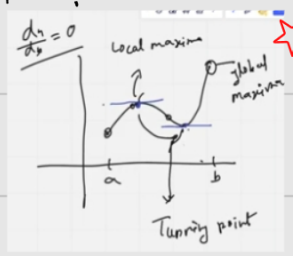
$\omega\left(\frac{\pi}{4\sqrt{3}}\right) = -0.186$

$\omega\left(-\frac{\pi}{4\sqrt{3}}\right) = 0.186$

Check { $\omega(-1) = 0.1383$, $\omega(1) = -0.1383$

$$\therefore |f(x) - P_2(x)| \le \left|\frac{0.916}{6} \times 0.3831\right| \le 0.0537$$

$\therefore$ Max^m error $= 0.0537$



we're to check for the interval values too in case they have a greater maximum value

we'll only deal with the absolute value

Polynomial Interpolation - 5

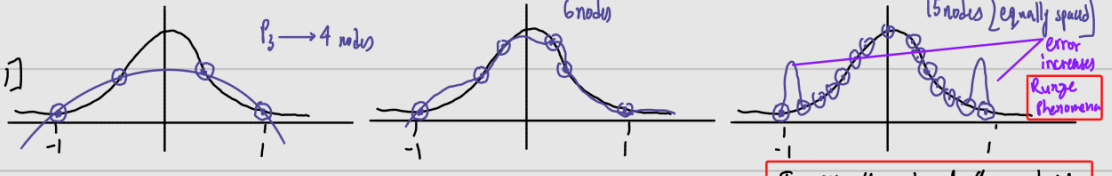
02/03/2025
Sunday

Convergence

nodes ↑ error ↓ → Not always true

$|f(x) - P_n(x)|$ $\therefore$ nodes $\infty \to$ error $= 0$

eg $f(x) = \frac{1}{1+25x^2}$, Interval $[-1,1]$



Increasing the nodes at those endpoints will decrease the errors.

Runge Phenomena

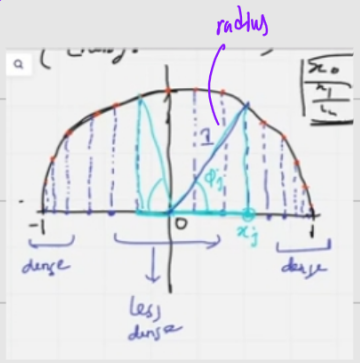
① Depends on the function (Mainly symmetric function) Runge function

② Depends on the nodes (Equally spaced)

Solution

① Piecewise Interpolation
- ↳ take smaller intervals and interpolate
- ↳ add them up.

② Non-equidistant nodes (Chebyshev's nodes) (instead of taking randomly)
- ↳ Draw a semi-circle
- ↳ Put a no. of equidistant points and identify the corresponding nodes.
- ↳ Observation: It appears that the nodes are more dense around the interval points
- ↳ Draw triangles with a perpendicular line joining the nodes with their corresponding points.
- ↳ Chebyshev relates the angles with the corresponding nodes:



$$\varphi_j = \frac{(2j+1)\pi}{2(n+1)}$$

$\cos\varphi_j = \frac{x_j}{1}$; radius $= 1$

$\Rightarrow x_j = \cos\varphi_j$

$\therefore x_j = \cos\frac{(2j+1)\pi}{2(n+1)}$

Runge function

eg (i) $f(x) = \frac{1}{1+25x^2}$ $[-1,1]$, $n=3$ Find the nodes

most of the times this interval is given

$= n=3$, so no. of nodes $= 4$ # $x_0 = \cos\frac{(2\times0+1)\pi}{2(n+1)} = \cos\pi/8$, $x_1 = \cos 3\pi/8$, $x_2 = \cos 5\pi/8$, $x_3 = \cos 7\pi/8$

☆ (ii) Using the nodes, define the Lagrange polynomial / Newton's Polynomial

Polynomial Interpolation - 6

Hermite Interpolation (aka Derivative Conditions)

Earlier: For n+1 nodes; $[x_0, x_1, x_2 \dots x_n]$ — n degree

has n+1 functions $[f(x_0), f(x_1) \dots f(x_n)]$ } n+1 conditions

hermite's idea: has n+1 first derivatives $[f'(x_0), f'(x_1) \dots f'(x_n)]$ → n+1 conditions

Total 2n+2 Conditions

n+1 nodes / conditions → degree n → Using Lagrange: n+1 nodes give n degree: P_n

2n+2 " n → degree: $(2n+2)-1 = 2n+1$ → Using Hermite: n+1 nodes give 2n+1 degree: P_{2n+1} → higher degree polynomial → better accuracy

eg given: 4 nodes

(i) $n = 3$

(ii) degree of Polynomial after using Lagrange/Newton: $P_n = P_3$

(iii) Using Hermite interpolation: $P_{2n+1} = P_{2\times3+1} = P_7$

Hermite Interpolation Theorem: Given nodes $(x_0, x_1, \dots x_n)$ and $f(x_0), f(x_1), \dots f(x_n)$ and $f'(x_0), f'(x_1), \dots f'(x_n)$ → then there exists an unique polynomial $P_{2n+1}(x)$

Lagrange: $P_n(x) = \sum_{k=0}^{n} f(x_k) l_k(x)$

Hermite: $P_{2n+1}(x) = \sum_{k=0}^{n} f(x_k) h_k(x) + f'(x_k) \hat{h}_k(x)$

$$l_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \left(\frac{x-x_j}{x_k-x_j}\right) = \left(\frac{x-x_0}{x_k-x_0}\right)\cdot\left(\frac{x-x_1}{x_k-x_1}\right)\left(\frac{x-x_2}{x_k-x_2}\right)\cdots\left(\frac{x-x_n}{x_k-x_n}\right)$$

eg nodes (x_0, x_1)

$\Rightarrow n=1 \quad P_{2n+1} = P_{2\times1+1} = P_3 = \left[f(x_0)h_0(x) + f'(x_0)\hat{h}_0(x)\right] + \left[f(x_1)h_1(x) + f'(x_1)\hat{h}_1(x)\right]$

$$h_k(x) = \left[1 - 2(x-x_k)l_k'(x_k)\right]\{l_k(x)\}^2 \qquad \hat{h}_k(x) = (x-x_k)\{l_k(x)\}^2$$

eg $f(x) = \sin x$ nodes $\{0, \pi/2\}$

Any $n=1$ $f(0)=0$ $f'(x)=\cos x$ $f'(0)=1$ $f(\pi/2)=1$ $f'(1)=0$

$$P_{2\times1+1}(x) = \{0\times h_0(x) + 1\times\hat{h}_0(x)\} + \{1\times h_1(x) + 0\times\hat{h}_1(x)\}$$
$$= \{(x-0)l_0^2(x)\} + \{1-2(x-\pi/2)l_1'(\pi/2)l_1^2(x)\}$$
$$= x\left(\frac{1-2x}{\pi}\right)^2 + 1 - 2(x-\pi/2)\cdot\frac{2}{\pi}\left(\frac{4x^2}{\pi^2}\right)$$

$$l_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{n} \left(\frac{x-x_j}{x_k-x_j}\right)$$

$$l_0(x) = \frac{x-x_1}{x_0-x_1} = \frac{x-\pi/2}{0-\pi/2} = 1 - \frac{2x}{\pi}$$

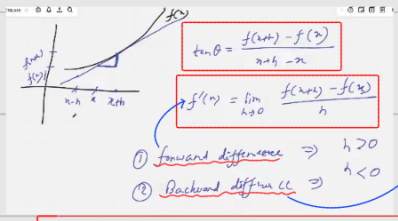
$$l_1(x) = \frac{x-x_0}{x_1-x_0} = \frac{x-0}{\pi/2-0} = \frac{2x}{\pi} \qquad l_0^2(x) = \left(1-\frac{2x}{\pi}\right)^2$$

$$l_1'(x) = \frac{2}{\pi} \qquad l_1^2(x) = \frac{4x^2}{\pi^2}$$

$$\therefore l_1'(\pi/2) = \frac{2}{\pi}$$

Differentiation-1

22/03/2025 Saturday



$$f'(x) = \frac{f(x+h)-f(x)}{h}$$

h↑ Accuracy↓

$f(x) = P(x) + \text{Error}$

$$f(x) = f(x_0)\left(\frac{x-x_1}{x_0-x_1}\right) + f(x_1)\left(\frac{x-x_0}{x_1-x_0}\right) + \frac{f''(\xi)}{2!}(x-x_0)(x-x_1)$$

Lagrange — u — Cauchy — v

$$f'(x) = \frac{f(x_0)}{x_0-x_1} + \frac{f(x_1)}{x_1-x_0} + \frac{f''(\xi)}{2}\frac{d}{dx}(\xi)(x-x_0)(x-x_1) + \frac{f'(\xi)}{2}(2x-x_0-x_1)$$

$$f'(x_0) = \frac{f(x_0)}{x_0-x_1} + \frac{f(x_1)}{x_1-x_0} + \frac{f''(\xi)}{2}(x_0-x_1) \quad [x = x_0]$$

$$f'(x_0) = -\frac{f(x_0)}{h} + \frac{f(x_0+h)}{h} - \frac{f''(\xi)}{2}(h) \quad [x_1 = x_0+h]$$

Truncation error → in case of forward differentiation → Error rate $\propto h$ ☆ → Same for backward diff

$$f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h} - \frac{f^{(2)}(\xi)}{2}(h)$$

Q) Find forward difference and Truncation error.

$f(x) = \ln(x)$, $x_0 = 2$

$h = 1, 0.1, 0.01, 0.001$

Ans: $f'(x) = \frac{1}{x}$

$f'(2) = \frac{1}{2} = 0.5$

h	Forward difference	Truncation Error
1	0.4054651	0.0794535
0.1	0.4879201	0.0019208
0.01	- - - -	- - -
0.001	- - - -	- - -

h↑ Error↓

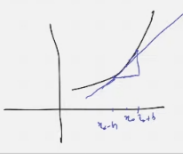
$$\frac{f(x+h)-f(x)}{h} \to f'(x)$$

$$= \frac{\ln(2+1)-\ln(2)}{1} = 0.4054651$$

If h values are not given, choose it yourself! Small values

In case of backward difference, error will be -ve

Central difference



$$f'(x_0) = \lim_{h\to 0}\frac{f(x_0+h)-f(x_0-h)}{2h}$$

$x-h$	x_0	x_0
x_0	x_0+h	x_1
x_0+h	x_0+2h	x_2

$$f(x) = f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} + \frac{f'''(\xi)}{6}(x-x_0)(x-x_1)(x-x_2)$$

$$f'(x_0) = \frac{f(x_0)-f(x_0)}{2h} - \frac{f'''(\xi)}{6}(h)^2$$

Error

or $$f'(x_0) = \frac{f(x_0+h)-f(x_0-h)}{2h} + \frac{f^{(3)}(\xi)}{6}(h^2)$$ → Error $\propto h^2$ ☆

error

$$f'(x) = \frac{f(x+h)-f(x-h)}{2h}$$

$$= \frac{\ln(x+h)-\ln(x-h)}{2h}$$

$$= \frac{\ln(2+1)-\ln(2-1)}{2\cdot 1}$$

Q) Find central difference and Truncation error.

$f(x) = \ln(x)$, $x_0 = 2$

$h = 1, 0.1, 0.01, 0.001$

$f'(x) = \frac{1}{x}$

$f'(2) = \frac{1}{2} = 0.5$

h	Central difference	(0.5 − central diff) Truncation error
1	0.5493061	−0.0493061
0.1	1	1
0.01	1	1
0.001	1	1

-ve

Differentiation-2

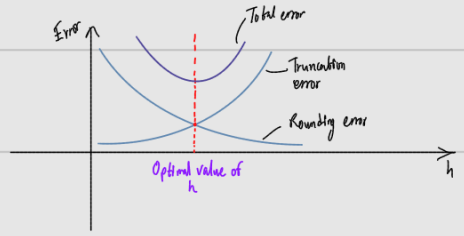
In $f(x+h)$, $f(x-h)$, h/h^2 rounding error occurs

$$\text{Error} = \text{Truncation error} + \text{Rounding error} = -\frac{f'''(\xi)}{6}h^2 + \epsilon_m\frac{f(x+h)-f(x-h)}{2h}$$

machine epsilon

$$\text{Error} \le \left|\frac{f'''(\xi)}{6}h^2\right| + \epsilon_m\frac{|f(x+h)-f(x-h)|}{2h}$$

Error $\propto h^2$ — Error $\propto \frac{1}{h}$



Central: $\frac{f'''(\xi)}{6}h^2 \Rightarrow$ Error is of order $h^2 = O(h^2)$

forward: $\frac{f''(\xi)}{2}h \Rightarrow O(h)$

Richardson Extrapolation

works for central difference Only

Derivation may come ☆

forward $\frac{f(x+h)-f(x)}{h} - \frac{f''(\xi)}{2}h$

central $\frac{f(x+h)-f(x-h)}{2h} - \frac{f'''(\xi)}{6}h^2$

$\frac{f(x+2h)-f(x-2h)}{2h} - \frac{f'''(\xi)}{6}h^2$

$$D_h = \frac{f(x+h)-f(x-h)}{2h}$$ 2nd order approximation

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \frac{f^4(x)}{4!}h^4 + \frac{f^5(x)}{5!}h^5 + O(h^6)$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 + \frac{f^4(x)}{4!}h^4 - \frac{f^5(x)}{5!}h^5 + O(h^6)$$

$$\#h = \frac{f(x+h)-f(x-h)}{2h} = \frac{1}{2h}\left(2f'(x)h + 2\frac{f'''(x)}{3!}h^3 + 2\frac{f^5(x)}{5!}h^5 + O(h^7)\right)$$

$$D_h = f'(x) + \frac{f'''(x)}{3!}h^2 + \frac{f^5(x)}{5!}h^4 + \theta(h^6)$$

Truncation error

when h↓, h^2 decrease → $O(h^4)$ — $O(h^2)$

less than h^2

$$D_h = f'(x) + \frac{f'''(x)}{3!}h^2 + \frac{f^5(x)}{5!}h^4 + O(h^6) \quad -(i)$$

$$D_{\frac{h}{2}} = f'(x) + \frac{f'''(x)}{3!}\frac{h^2}{4} + \frac{f^5(x)}{5!}\frac{h^4}{16} + O(h^6) \quad -(ii)$$

$4\cdot D_{\frac{h}{2}} - D_h = 3f'(x) + 0 + \left(\frac{1}{4}-1\right)\frac{f^5(x)}{5!}h^4 + O(h^6)$

$\frac{4D_{\frac{h}{2}} - D_h}{3} = f'(x) - \frac{1}{4}\frac{f^5(x)}{5!}h^4 + O(h^6) = D_h^{(1)}$

$$D_h^{(1)} = \frac{2^2 D_{h/2} - D_h}{2^2-1} = f'(x) + O(h^4)$$

$$D_h^{(2)} = \frac{2^4 D_{h/2}^{(1)} - D_h^{(1)}}{2^4-1}$$

$D_h^{(2)}$

$D_h^{(1)}$

$$D_h^{(1)} = f'(x) - \frac{1}{4}\frac{f^5(x)}{5!}h^4 + O(h^6) \quad -(i)$$

$$D_{\frac{h}{2}}^{(1)} = f'(x) - \frac{1}{4}\frac{f^5(x)}{5!}\frac{h^4}{16} + O(h^6) \quad -(ii)$$

$16D_{\frac{h}{2}}^{(1)} - D_h^{(1)} = 15f'(x) + O(h^6)$

$\frac{16D_{\frac{h}{2}}^{(1)} - D_h^{(1)}}{15} = f'(x) + O(h^6) = D_h^{(2)}$

We could have applied this Richardson extrapolation procedure without knowing the coefficients of the error series. If we have some general order-n approximation

$$D_h = f'(x) + C(h^n) + O(h^{n+1})$$

then we can always evaluate it with $\frac{h}{2}$ to get ☆

$$D_{\frac{h}{2}} = f'(x) + C\left(\frac{h}{2}\right)^n + O(h^{n+1})$$

and then eliminate the h^n term to get a new approximation

$$D_h^{(1)} = \frac{2^n D_{\frac{h}{2}} - D_h}{2^n - 1} = f'(x) + O(h^{n+1})$$

Final

Chapter 4: Non-Linear equations

Fixed Point Iteration - 1

28/03/2025
Friday

Solving non-Linear eqn

eg $x^3+2x^2+3x+5=0$

$x = \square, \square, \square$

Finding roots

eg $f(x)=x^2-4$

$f(x)=0$

$\therefore x=\pm 2$ (roots)

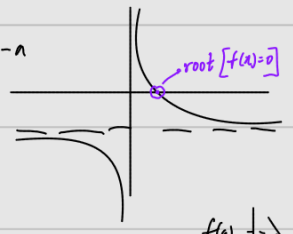
$\downarrow$

x_*

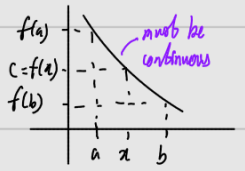
eg $f(x)=\frac{1}{x}-a$

$f(x)=0$

$x=\frac{1}{a}$



**** This is how we find the roots but Computer uses different Algorithms.



$c \in [f(a), f(b)]$

$x \in [a, b]$

1] Interval Bisection/Bisection Method: Based on Intermediate Value Theorem

(4.1: Out of Syllabus but impt for Lab) ↳ Finding roots of non-linear eqn

Q) $f(x)=x-0.5$, $a_0=1$, $b_0=3$

$f(a_0)=\frac{1}{1}-0.5=0.5$, $f(b_0)=-0.17$

a_k
b_k
m_k

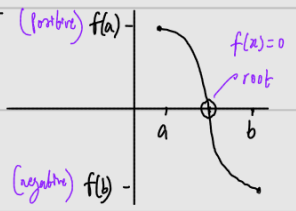
$f(a)\cdot f(b)<0 \Rightarrow$ At least one root exists

if $f(a)\cdot f(b)>0 \Rightarrow$ No root exists

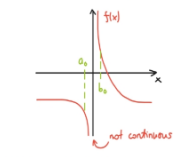
midpoints, $m_0=\frac{a_0+b_0}{2}=\frac{1+3}{2}=2$

$f(m_0)=f(2)=\frac{1}{2}-0.5=0$

$m_0=x_*=2 \Rightarrow$ root //



Example → Same example with initial interval $[-0.5, 0.5]$.



In this case $f(a_0)f(b_0) < 0$, but there is no root in the interval.

1st Iteration ($k=0$)

Q) $f(x)=\frac{1}{x}-0.5$, $a_0=1.5$, $b_0=3$

$f(a_0)=0.17$, $f(b_0)=-0.17$

$\therefore f(a_0)\cdot f(b_0)<0 \Rightarrow$ root exists //

$m_0=\frac{1.5+3}{2}=2.25$

$f(m_1)=\frac{1}{2.25}-0.5=-0.056$

as it is -ve, update b_0

2nd Iteration

$a_1=1.5$ $b_1=m_0=2.25$ $f(a_1)\cdot f(b_1)<0$ ☆

$f(a_1)=0.17$ $f(b_1)=-0.056$ $m_1=\frac{a_1+b_1}{2}=1.875$

$\rightarrow f(m_1)=0.033 \rightarrow$ +ve, update a

3rd Iteration

$a_2=m_1=1.875$, $b_2=2.25$

$m_2=\frac{a_2+b_2}{2}=2.0625$

$\rightarrow f(m_2)=-0.01525 \rightarrow$ -ve, update b_2

must check $f(a_2)\cdot f(b_2)<0$

☆ Taking m_0 close to zero ☆

↑

Convergence

↓

slow

no. of iterations will be given in the question

Use Table

	a	b	m	f(a)	f(b)	f(m)
k=0						
k=1						
k=2						
⋮						

*** The rate of convergence is steady, so we can pre-determine how many iterations will be needed to converge to a given accuracy. After k iterations, the interval has length

$$|b_k - a_k| = \frac{|b_0 - a_0|}{2^k}. \quad (4.2)$$

so the error in the mid-point satisfies

$$|m_k - x_*| \le \frac{|b_0 - a_0|}{2^{k+1}}. \quad (4.3)$$

In order for $|m_k - x_*| \le \delta$ we need n iterations, where

$$\frac{|b_0 - a_0|}{2^{n+1}} \le \delta \implies \log|b_0 - a_0| - (n+1)\log(2) \le \log(\delta) \implies n \ge \frac{\log|b_0 - a_0| - \log(\delta)}{\log(2)} - 1. \quad (4.4)$$

Example → With $a_0 = 1.5$, $b_0 = 3$, as in the above example, then for $\delta = \epsilon_M = 1.1 \times 10^{-16}$ we would need

$$n \ge \frac{\log(1.5) - \log(1.1 \times 10^{-16})}{\log(2)} - 1 \implies n \ge 53 \text{ iterations.}$$

▸ This convergence is pretty slow, but the method has the advantage of being very robust (i.e., use it if all else fails...). It has the more serious disadvantage of only working in one dimension.

Advantage: Method is robust

Disadvantage: The convergence is pretty slow

Only works in One dimension

$a_0=1.5$, $b_0=3$, $|b_0-a_0|=1.5$

$b_1=2.25$, $a_1=1.5$, $|b_1-a_1|=0.75$

$\therefore |b_1-a_1|=\frac{|b_0-a_0|}{2}$

$|b_2-a_2|=\frac{|b_1-a_1|}{2}=\frac{\frac{|b_0-a_0|}{2}}{2}=\frac{|b_0-a_0|}{2\cdot 2}$

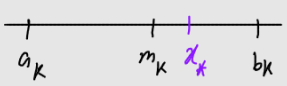
$|b_3-a_3|=\frac{|b_0-a_0|}{2\cdot 2\cdot 2}$

$\therefore |b_k-a_k|=\frac{|b_0-a_0|}{2^k}$, k = no. of iteration +1

As m_k is never equal to x_*,

$|m_k - x_*|$ = Error

actual error



$\therefore |m_k-x_*| \le \frac{|b_k-a_k|}{2}$

$\Rightarrow |m_k-x_*| \le \frac{\frac{|b_0-a_0|}{2^k}}{2}$

$\therefore |m_k-x_*| \le \frac{|b_0-a_0|}{2^{k+1}}$

Actual error — max^m possible error

$\frac{|b_0-a_0|}{2^{k+1}} \le \delta$

$\Rightarrow \ln\left|\frac{|b_0-a_0|}{2^{k+1}}\right| \le \ln\delta$

$\Rightarrow \ln|b_0-a_0| - \ln 2^{k+1} \le \ln\delta$

$\Rightarrow -\ln 2^{k+1} \le \ln\delta - \ln|b_0-a_0|$

$\Rightarrow \ln 2^{k+1} \ge \ln|b_0-a_0| - \ln\delta$

$(k+1)\ln 2 \ge \ln|b_0-a_0| - \ln\delta$

$\therefore k \ge \frac{\ln|b_0-a_0| - \ln\delta}{\ln 2} - 1$

eg $\delta=0.00001$, $a_0=1.5$, $b_0=3$

$k \ge \frac{\ln|3-1.5| - \ln(0.00001)}{\ln 2} - 1$

$k \ge 16.19$

$\therefore k=17$ (18 iterations)

eg $\delta = \epsilon_m$ (machine epsilon) $= 1.1\times10^{-16}$, $a_0=1.5$, $b_0=3$

$k \ge \frac{\ln|3-1.5| - \ln(1.1\times10^{-16})}{\ln 2}$

$k \ge 52.6$

$\therefore k=53$ //

Bisection Method Disadvantages

↳ Convergence is slow

↳ Multiple roots can't be found

→ Hence

Fixed point iteration

↳ Convergence is faster

↳ Can find multiple roots

$f(x)=0$

$g(x)=x \Rightarrow$ fixed point (x,y)

(4.2: Fixed point iteration)

Fixed Point Iteration - 2

13/04/2025
Sunday

e.g. $f(x) = x^2 - 2x - 3$

using middle term, $x_* = 3, x_* = -1$

$g(x) = x$ (will be given)

$x^2 - 2x - 3 = 0$

$x^2 = 2x + 3$

$x = \sqrt{2x+3} = g(x)$, given $x_0 = 0$ (initial value)

$g(0) = \sqrt{3} = 1.7321$ $x_0 = 0$

$g(\sqrt{3}) = 2.5425$ $x_1 = \sqrt{3}$

$g(2.5425) = 2.8934$ $x_2 = 2.5425$

$g(2.84) = 2.94$ $x_3 = 2.8434$

4 iterations

$g(x) = \sqrt{2x+3}$ $\Rightarrow$ requires 33 iterations

$g(x) = \frac{3}{x-2}$ $\Rightarrow$ 33 "

$g(x) = \frac{x^2-3}{2}$ $\Rightarrow$

$g(x) = x^2 - x - 3$ $\Rightarrow \infty$

$g(x) = \frac{x^2-3}{2x-2}$ $\Rightarrow$ 5 iterations

Computer chooses the best one; using Contraction Method

Contraction Mapping

Contraction mapping

$g(x) = x^2$

$g(1) = 1$

$g(2) = 4$

Not contraction map (range increases)

eg. $g(x) = \sqrt{2x}$

$g(1) = 1$

$g(2) = 1.4$

range decreases $\Rightarrow$ Contraction mapping

$|g(x) - g(y)| \leq \lambda |x - y| \Rightarrow$ contraction map

$\lambda < 1 \Rightarrow$ Contraction map

$\lambda \propto \frac{1}{\text{Convergence Rate}}$ ☆

Contraction Mapping is when $\lambda = 1$ ☆

Bisection Method: $\lambda = \frac{1}{2}$

$|g(x) - g(y)| \leq \lambda |x - y|$

$|g(x) - g(y)| \leq |g'(\xi)| \, |x - y|$

eg. $g(x) = \sqrt{2x+3}$

$g'(x) = \frac{1}{2}(2x+3)^{-\frac{1}{2}} \cdot 2$

$= \frac{1}{\sqrt{2x+3}}$

$|\lambda| < 1$

$g'(x) < 1$

$\frac{1}{\sqrt{2x+3}} < 1$ (largest value)

$\Rightarrow 1 < 2x + 3$

$\Rightarrow -2 < 2x$

$\therefore x > -1$

$\therefore$ Contraction mapping of function $g(x)$ is possible if $x > -1$

eg. $f(x) = \sqrt{2x+3}$, $x = \frac{1}{2}$, $y = 4$

$g(-\frac{1}{2}) = \sqrt{2}$

$g(y) = g(4) = \sqrt{11}$

$\therefore |g(x) - g(y)| \quad |x - y|$

$\therefore 3.32 < 4.5$

$\therefore$ Contraction mapping possible

☆ if x, y given but not λ value, then contraction mapping is possible if $|g(x) - g(y)| < |x - y|$

(1) $g(x) = \sqrt{2x+3}$, $x_0 = 0$, $x_* \to 3$

$x_1 = g(x_0)$

	x_k	$\lvert x_* - x_k \rvert$	$\lvert x_k - x_{k-1} \rvert / \lvert x_{k-1} - x_{k-2} \rvert$
$x_0 \to$	0	3	Can't Calculate
$x_1 \to$	$\sqrt{3}$	$1.2675 (3 - \sqrt{3})$	0.4226
x_2	$g(\sqrt{3})$	0.4575	0.3608
x_3			0.3423
x_4			0.3365
			0.3339

getting towards $0.333 = \frac{1}{3}$

$\therefore \lambda = \frac{1}{3}$

(2) $g(x) = \frac{x+5}{2x-2}$, $x_0 = 0$

0.5

0.1

0.012

0.0001492...

$\lambda = 0$

high rate of convergence

☆ if $\lambda = 0 \Rightarrow$ Super Linear Convergence

$\lambda = \frac{1}{3} / \frac{1}{2} / \ldots \Rightarrow$ Linear Convergence

☆ if $\lambda > 1 \Rightarrow$ Divergence. Hence, we can't find root

20/04/2025
Sunday

Newton's Method & Aitken Acceleration 1

So Far, (1) $g(x) = \sqrt{2x+3}$, $x_* = 3$

$g'(x) = \frac{1}{\sqrt{2x+3}}$

$g'(x_*) = g'(3) = \frac{1}{\sqrt{2x3+3}} = \frac{1}{3} = \lambda$ (Linear Convergence)

(2) $g(x) = \frac{x^2+3}{2x-2}$, $x_* = -1$

$g'(x) = \frac{x^2 - 2x - 3}{2(x-1)^2}$

$g'(-1) = 0 = \lambda$ (Super Linear Convergence)

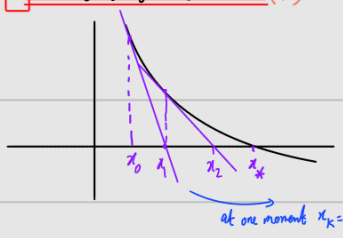
Disadvantage of Bisection Method:

- Convergence rate is $\frac{1}{2}$ which is very low

Thus we used fixed point iteration to reduce value of λ (less than $\frac{1}{2}$) but not always zero.

*** Newton's Method convergence rate is always zero

Newton's Method (4.4) : This is a particular fixed point iteration that is very widely used bcos it usually converges Superlinearly.



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

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

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

$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}}$

λ tends to zero

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

$g'(x) = \frac{f(x) \times f''(x)}{(f'(x))^2}$, $\boxed{f(x_*) = 0}$

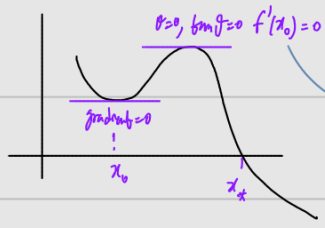
$g'(x_*) = \frac{f(x_*) \times f''(x_*)}{(f'(x_*))^2} = 0$

$g'(x_*) = 0 \longrightarrow \lambda$

Derivation: $m = f'(x_k) = \frac{0 - f(x_k)}{x_{k+1} - x_k}$

$\therefore x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

Sometimes, Newton's Method doesn't give a soln



$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ → undefined

No soln
Infinite Loop

Disadvantage of Newton's Method ☆

> ▸ In fact, Newton only applied the method to polynomial equations, and without using calculus. The general form using derivatives ("fluxions") was first published by Thomas Simpson in 1740. [See "Historical Development of the Newton-Raphson Method" by T.J. Ypma, *SIAM Review* **37**, 531 (1995).]

eg $f(x) = \frac{1}{x} - a$, $x_* = \frac{1}{a}$, $a > 0$

$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

$= x_k - \frac{(\frac{1}{x_k} - a)}{-\frac{1}{x_k^2}}$

$\vdots$

$x_{k+1} = 2x_k - a x_k^2$

given $a = 0.5$, $x_0 = 1$

$\therefore x_* = \frac{1}{a} = \frac{1}{0.5} = 2$

$2(1) - 0.5(1)^2 = 1.5$

x_k	$\lvert x_* - x_k \rvert$	$\lvert x_* - x_k \rvert / \lvert x_k - x_{k-1} \rvert = \lambda$
$x_0 = 1$	1	—
$x_1 = 1.5$	0.5	0.5
$x_2 = 1.875$	0.125	0.25
$x_3 = 1.9961$	0.00781	0.06248
$x_4 = 1.99997$	0.00003	0.00384
$x_5 = 2$	0	$0 \longrightarrow \lambda$

eg $f(x) = x^3 - 2x + 2$, $x_* \simeq -1.7693$, $x_0 = 0$

$f'(x) = 3x^2 - 2$

$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = \frac{x_k^3 - 2x_k + 2}{3(x_k)^2 - 2}$

$x_0 = 0$ $x_1 = 1$
$x_1 = 1$ $x_2 = 0$
$x_2 = 0$ $x_3 = 1$
$x_3 = 1$ $x_4 = 0$
$x_5 = 1$
$x_6 = 0$

Infinite loop because of $x_0 = 0$
Hence No soln
It'll never converge //

Disadvantage of Newton's Method

(4.5 : Out of syllabus)

Solving two Equations

$f_1(x,y) = xy - y^3 - 1 = 0$ $x_0 = 2$,

$f_2(x,y) = x^2 y + y - 5 = 0$ $y_0 = 3$

$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} \frac{\delta f_1}{\delta x} & \frac{\delta f_1}{\delta y} \\ \frac{\delta f_2}{\delta x} & \frac{\delta f_2}{\delta y} \end{pmatrix}^{-1} \times \begin{pmatrix} f_1(x_k, y_k) \\ f_2(x_k, y_k) \end{pmatrix}$

Jacobean Matrix

$\frac{\delta f_1}{\delta x} = y = 3$, $\frac{\delta f_1}{\delta y} = x - 3y^2 = 2 - 3(3)^2 = -25$

$\frac{\delta f_2}{\delta x} = 2xy = 2 \cdot 2 \cdot 3 = 12$, $\frac{\delta f_2}{\delta y} = x^2 + 1 = 2^2 + 1 = 5$

$\therefore \begin{pmatrix} 3 & -25 \\ 12 & 5 \end{pmatrix}^{-1} = \begin{pmatrix} 1/63 & 5/63 \\ -4/105 & 1/105 \end{pmatrix}$

$\therefore \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 1/63 & 5/63 \\ -4/105 & 1/105 \end{pmatrix} \times \begin{pmatrix} -22 \\ 10 \end{pmatrix}$

$= \begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 4/7 \\ 14/15 \end{pmatrix}$

$\therefore \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 14/7 \\ 31/15 \end{pmatrix}$ //

$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1.54 \\ 1.97 \end{pmatrix}$

$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1.78 \\ 1.97 \end{pmatrix}$

$\vdots$

$\begin{pmatrix} x_5 \\ y_5 \end{pmatrix} = \begin{pmatrix} 1.999 \\ 1.001 \end{pmatrix}$ — Approaches $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ //

Newton's Method & Aitken Acceleration-2

21/04/2025
Monday

g) $f(x) = x^2 - 2xe^{-x} + e^{-2x}$, $x_0 = 1$ with $\varepsilon = 10^{-5}$ $f(x_k) < 10^{-5}$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$x_{k+1} = x_k - \frac{x_k^2 - 2x_k e^{-x_k} + e^{-2x_k}}{2x_k - 2e^{-x_k} + 2x_k e^{-x_k} + 2e^{-2x_k}}$$

x_k	$f(x_k)$	is $f(x_k) < 10^{-5}$?
x_0 1	0.399576	NO
x_1 0.768591	0.093202	NO
x_2 0.664990	0.002253	NO
$\vdots$		
$x_8 = 0.56625$	0.5×10^{-5}	YES

Use Calculator Tricks

*** Sometimes convergence rate is very low, hence it needs to be increased

Aitken Acceleration → Increases convergence Rate

(9.6: Out of syllabus)

eg, $f(x) = \frac{1}{x} - 0.5$, $x_0 = 2$

$g(x) = x + \frac{1}{16}\left(\frac{1}{x} - 0.5\right)$ ($f(x)$)

$g'(x) = 1 + \frac{1}{16}\left(-\frac{1}{x^2}\right)$

$\lambda = g'(x_*) = g'(2) = 0.984375 \Rightarrow$ ✓

*** To solve this problem it'll take x_{181} iterations

eg $f(x) = \frac{1}{x} - 0.5$, $x_0 = 1.5$

$g(x) = x + \frac{1}{16}\left(\frac{1}{x} - 0.5\right)$

$$\hat{x}_{k+2} = x_k - \frac{(x_{k+1} - x_k)^2}{x_{k+2} - 2x_{k+1} + x_k}$$

min^m two iterations needed to find this

$x_1 = g(x_0) = 1.510417$

$x_2 = g(x_1) = 1.520596$

$$\hat{x}_2 = x_0 - \frac{(x_1 - x_0)^2}{x_2 - 2x_1 + x_0} = 1.877664$$

$x_3 = g(\hat{x}_2) = 1.879691$

$x_4 = g(x_3) = 1.881642$

$$\hat{x}_4 = x_2 - \frac{(x_3 - x_2)^2}{x_4 - 2x_3 + \hat{x}_2} = 1.992634$$

$\vdots$

$\hat{x}_8 = 2.00000$

Using fixed point iteration, it would have taken 819 iterations, here it only took eight

Disadvantage of Newton's Method

*** Newton's Method has high computation cost thus to reduce cost, $f'(x)$ is replaced by an approximate function $g(x)$

$$x_{k+1} = x_k - \frac{f(x)}{f'(x)}$$

$f'(x) \approx g(x)$: Quasi-Newton's Method (9.7)
- Secant method
- Steffensen's method (out of syllabus)

$x_{k+1} = x_k - \frac{f(x)}{f'(x)}$, Backward Difference $= \frac{f(x) - f(x-h)}{h}$, $g_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$

$$\Rightarrow x_{k+1} = x_k - \frac{f(x_k)}{g_k} = x_k - \frac{f(x_k)\,[x_k - x_{k-1}]}{f(x_k) - f(x_{k-1})}$$

No $f'(x_k)$ hence less cost

☆ Secant Method requires two starting points ☆

eg $f(x) = \frac{1}{x} - 0.5$, $x_* = 2$, $x_0 = 0.25$, $x_1 = 0.5$

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)} = 0.6875$$

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

$\vdots$

$x_8 = 1.99916$

$\vdots$

$x_{11} = 2$

Use Calculator Tricks

$|x_k - x_*| / |x_* - x_{k-1}|$

$= 0.75$

$= 0.65625$

$\vdots$

$= 0.05194$

$\vdots$

$\lambda \to 0$ [Super Linear Convergence]

Using Newton's Method, λ is always zero

As Secant Method is part of / improved version of Newton's Method, hence here λ is also zero which is Super Linear Convergence

Gaussian Elimination method and LU decomposition - 1

23/04/2025
Wednesday

$$x_1 - 2x_2 + x_3 = 0$$
$$x_1 - 2x_2 + 2x_3 = 4$$
$$2x_1 + 12x_2 - 2x_3 = 4$$

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & -2 & 2 \\ 2 & 12 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 4 \end{pmatrix}$$

A x b

$A \cdot x = b$

$x = A^{-1} b$

Don't mix up with Vandermonde's Matrix. That solves Polynomial and now we're solving Linear System. ☆

Unique sol^n exists if:

(1) A square matrix

(2) $\det(A) \neq 0$ [non singular matrix]

Finding inverse matrix is of computationally high cost hence computer uses a different approach to solve the Linear system by using:

In general, any lower triangular system $Lx = b$ can be solved by *forward substitution*

$$x_j = \frac{b_j - \sum_{k=1}^{j-1} l_{jk} x_k}{l_{jj}}, \quad j = 1, \ldots, n. \tag{5.5}$$

Similarly, an *upper triangular* matrix U has the form

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{pmatrix}. \tag{5.6}$$

and an upper-triangular system $Ux = b$ may be solved by *backward substitution*

$$x_j = \frac{b_j - \sum_{k=j+1}^{n} u_{jk} x_k}{u_{jj}}, \quad j = n, \ldots, 1. \tag{5.7}$$

To estimate the computational cost of forward substitution, we can count the number of floating-point operations ($+, -, \times, \div$).

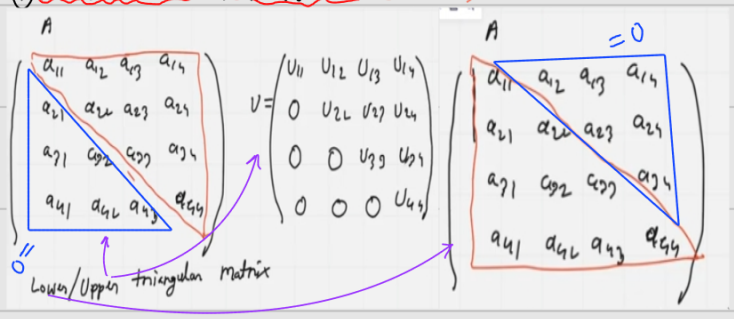
Example → Number of operations required for forward substitution.
Consider each x_j. We have

$j = 1 \rightarrow 1$ division
$j = 2 \rightarrow 1$ division + [1 subtraction + 1 multiplication]
$j = 3 \rightarrow 1$ division + $2 \times$[1 subtraction + 1 multiplication]
$\vdots$
$j = n \rightarrow 1$ division + $(n-1) \times$[1 subtraction + 1 multiplication]

So the total number of operations required is

$$\sum_{j=1}^{n} \left(1 + 2(j-1)\right) = 2\sum_{j=1}^{n} j - \sum_{j=1}^{n} 1 = n(n+1) - n = n^2.$$

(1) Gaussian Elimination: (5.1, 5.2)



Upper triangular matrix → Bottom up approach
Lower triangular matrix → Top down approach
$\} O(n^2)$

Computational Complexity of the algorithm

$A = n \times n$ Lower/Upper triangular matrix transformation $= O(n^2)$ //

Example → Number of operations required to find U.
Computing $A^{(k+1)}$ requires:

- $n - (k+1) + 1 = n - k$ divisions to compute m_{ik}.
- $(n-k)^2$ subtractions and the same number of multiplications to compute $a_{ij}^{(k+1)}$.

So in total $A^{(k+1)}$ requires $2(n-k)^2 + n - k$ operations. Overall, we need to compute $A^{(k+1)}$ for $k = 1, \ldots, n-1$, so the total number of operations is

$$N = \sum_{k=1}^{n-1} \left(2n^2 + n - (4n+1)k + 2k^2\right) = n(2n+1)\sum_{k=1}^{n-1} 1 - (4n+1)\sum_{k=1}^{n-1} k + 2\sum_{k=1}^{n-1} k^2.$$

Recalling that

$$\sum_{k=1}^{n} k = \tfrac{1}{2}n(n+1), \qquad \sum_{k=1}^{n} k^2 = \tfrac{1}{6}n(n+1)(2n+1),$$

we find

$$N = n(2n+1)(n-1) - \tfrac{1}{2}(4n+1)(n-1)n + \tfrac{1}{3}(n-1)n(2n-1) = \tfrac{2}{3}n^3 - \tfrac{1}{2}n^2 - \tfrac{1}{6}n.$$

So the number of operations required to find U is $O(n^3)$. ☆

▸ It is known that $O(n^3)$ is not optimal, and the best theoretical algorithm known for inverting a matrix takes $O(n^{2.3728639})$ operations (although this is not practically useful). But it remains an open conjecture that there exists an $O(n^{2+\epsilon})$ algorithm, for ϵ arbitrarily small.

Augmented Matrix

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 1 & -2 & 2 & 4 \\ 2 & 12 & -2 & 4 \end{array}\right)$$

make these 0

Row multiplier, $m_{21} = \frac{a_{21}}{a_{11}} = \frac{1}{1} = 1$

$R_2' \Rightarrow R_2 - R_1 \times m_{21}$

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -4 & 1 & 4 \\ 2 & 12 & -2 & 4 \end{array}\right)$$

$m_{31} = \frac{a_{31}}{a_{11}} = \frac{2}{1} = 2$

$R_3' \Rightarrow R_3 - m_{31} \times R_1$

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -4 & 1 & 4 \\ 0 & 8 & -4 & 4 \end{array}\right)$$

$m_{32} = \frac{a_{32}}{a_{22}} = \frac{8}{-4} = -2$

$R_3' \Rightarrow R_3 - R_2 \times m_{32}$

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 0 \\ 0 & -4 & 1 & 4 \\ 0 & 0 & -2 & 12 \end{array}\right)$$

Upper triangular matrix

Bottom up approach

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 12 \end{pmatrix}$$

$-2x_3 = 12$ # $-4x_2 + x_3 = 4$ # $x_1 = 11$ //

$x_3 = -6$ // $x_2 = -5/4$ //

LU Decomposition

$A = LXU$

$Ax = b$

$LUx = b$, $Ux = y$

$y = ?$

Gaussian Elimination method and LU decomposition - 2 (5.3)

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & -2 & 2 \\ 2 & 12 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 4 \end{pmatrix}$$

$A^{(1)}$ x b

$m_{21} = \frac{a_{21}}{a_{11}} = \frac{1}{1} = 1$

$m_{31} = \frac{a_{31}}{a_{11}} = \frac{2}{1} = 2$

Unit lower triangular matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ a_{21} & 1 & 0 \\ a_{31} & a_{32} & 1 \end{pmatrix}$$

Frobenious Matrix, $F^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$

$$A^{(2)} = F^{(1)} \times A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 1 & -2 & 2 \\ 2 & 12 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 8 & -4 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 8 & -4 \end{pmatrix}, \quad F^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}$$

$m_{32} = \frac{a_{32}}{a_{22}} = \frac{8}{-4} = -2$

$\therefore F^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$

Use calculator

$A^{(3)} = F^{(2)} \times A^{(2)}$

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 8 & -4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 0 & -2 \end{pmatrix} = U$$

$A^{(3)} = U$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix}$$

$\det(L) = \det(U)$

L: Lower, U: Upper of A

Both L & U of A gives same sol^n

$A^{(1)} = LXU$ $\quad Ax = b$

$LUx = b$, $Ux = y$

$Ly = b$

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 4 \end{pmatrix}$$

$y_1 = 0$, $y_2 = 4$, $y_3 = 12$

$$\therefore \begin{pmatrix} 1 & 2 & 1 \\ 0 & -4 & 1 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 12 \end{pmatrix}$$

$Ux = y$

$x_1 = 11$, $x_2 = -\frac{5}{4}$, $x_3 = -6$

Substitution Process

↳ $O(n^2)$

LU decomposition

LU decomposition offers several advantages over Gaussian elimination, particularly when solving multiple systems of linear equations with the same coefficient matrix.

If you're solving the same system for different $\boldsymbol{b}$ vectors, you don't need to redo the decomposition each time [***just reuse L and U***]. In contrast, in the Gaussian elimination method, if $\boldsymbol{b}$ changes, we need to restart row operations from the beginning.

Pivoting:

↳ Swapping row/column

↳ Reduces Rounding error

eg $\begin{pmatrix} 0 & 3 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$ → Hence swap the rows

$m_{21} = \frac{a_{21}}{a_{11}} = \frac{2}{0}$ (undefined)

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$ //

Least Square Approximation - 1

29/09/2025
Thursday

→ So far we used 3 unknowns, 3 equations. 3×3 Matrix

→ But Gaussian elimination won't work when no. of unknowns ≠ no. of equations. This is called Overdetermined System

$\vec{x} = (2,3,5)$ $\vec{x} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$, $x^T = [2\ 3\ 5]$ ← Transpose

magnitude/length/norm, $|\vec{x}| = \sqrt{2^2+3^2+5^2} = \sqrt{33}$

$|\vec{x}| = \sqrt{x^T x} = \sqrt{38} \Rightarrow \ell_2$-norm

$|\vec{x}| = \sqrt{\vec{x}\cdot\vec{x}} = \sqrt{(2,3,5)\cdot(2,3,5)} = \sqrt{4+9+25} = \sqrt{38}$ (same)

Dot product/inner product: $\vec{x}\cdot\vec{y} = |x|\cdot|y|\cos\theta = x^T y$

$\vec{x} = (2,3,5)^T \Rightarrow \ell_1$-norm $= |2|+|3|+|5| = 10$

Orthogonality (6.1)

$\vec{x}, \vec{y}$ $\vec{x}\cdot\vec{y} = 0$, $x^T y = 0$ ($\cos\theta = 0$, $\theta = 90°$)

Orthonormal
(1) $x^T y = 0 \Rightarrow x_1^T x_2 = 0$
(2) $x^T x = 1$, $y^T y = 1 \Rightarrow$ Norm Unity

$S = \{x_i \mid x_i \in \mathbb{R}^n,\ x_i^T x_j = \delta_{ij}\}$ (Dimension; Kronecker delta)

$\delta_{ij} = \begin{cases} 1, & i=j \rightarrow u^T u \text{ (normality)} \\ 0 & i \neq j \rightarrow u^T v \text{ (orthogonality)} \end{cases}$

Kronecker Delta

Theorem: $Q^{-1} = Q^T$, $QQ^T = Q^T Q = I$

eg. $S = \left\{\frac{1}{\sqrt{5}}(2,1)^T, \frac{1}{\sqrt{5}}(1,-2)^T\right\}$

$\vec{u} = \frac{1}{\sqrt{5}}(2,1)^T$ $\vec{v} = \frac{1}{\sqrt{5}}(1,-2)^T$

$u^T u = \vec{u}\cdot\vec{u} = \frac{1}{5}(4+1) = 1$

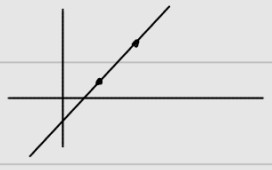
$v^T v = \vec{v}\cdot\vec{v} = \frac{1}{5}(1+(-2)^2) = 1$

$u^T v = \vec{u}\cdot\vec{v} = \frac{1}{5}(2\times1 - 1\times2) = 0$

$\therefore S$ is a orthonormal set $\mathbb{R}^2$

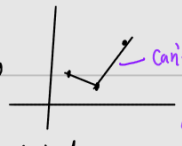
$P_n(x)$ Nodes = 2

$P_1(x) = a_0 + a_1 x$



Now for nodes = 3 and we wants $P_1(x)$

This is not possible, as there are two st. lines



Can't draw 1 degree st. line passing through all these points.

This is called Overdetermined System (more nodes available than needed)

related to Machine Learning

Solution: A straight line such that all points are equal dists from the line.

Eg) $n=1$, $x_0=-3$, $x_1=0$, $x_2=6$

$f(x_0)=0$, $f(x_1)=0$, $f(x_2)=2$

Ans $P_1(x) = a_0 + a_1 x$

$P_1(x_0) = a_0 + a_1 x_0 = f(x_0)$

$P_1(x_1) = a_0 + a_1 x_1 = f(x_1)$

$P_1(x_2) = a_0 + a_1 x_2 = f(x_2)$

Not square matrix, 3×2 hence No inverse

Matrix $\begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix}$

V a f

A x b

Produces a square matrix

$Ax = b$ (6.2)

$(A^T A)x = A^T b$

$\begin{pmatrix} 1 & -3 \\ 1 & 0 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$

$\Rightarrow \begin{pmatrix} 1 & 1 & 1 \\ -3 & 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & -3 \\ 1 & 0 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -3 & 0 & 6 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$

$\Rightarrow \begin{pmatrix} 3 & 3 \\ 3 & 45 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 12 \end{pmatrix}$

$\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 3 & 45 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 12 \end{pmatrix}$

$\det(A^T A) = 126 \neq 0$ hence has inverse

$\therefore \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \frac{1}{126} \begin{pmatrix} 45 & -3 \\ -3 & 3 \end{pmatrix} \begin{pmatrix} 2 \\ 12 \end{pmatrix} = \begin{pmatrix} 3/7 \\ 5/21 \end{pmatrix}$

$\therefore P_1(x) = a_0 + a_1 x$

$= 3/7 + 5/21\, x$

Theorem 6.2. *Let Q be an $m \times n$ matrix. The columns of Q form an orthonormal set iff $Q^T Q = I_n$.*

If $m = n$, then such a Q is called an *orthogonal matrix*. For $m \neq n$, it is just called a *matrix with orthonormal columns*.

Proof. Let $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ be the columns of Q. Then

$$Q^T Q = \begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{pmatrix} = \begin{pmatrix} \mathbf{q}_1^T\mathbf{q}_1 & \mathbf{q}_1^T\mathbf{q}_2 & \cdots & \mathbf{q}_1^T\mathbf{q}_n \\ \mathbf{q}_2^T\mathbf{q}_1 & \mathbf{q}_2^T\mathbf{q}_2 & \cdots & \mathbf{q}_2^T\mathbf{q}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_n^T\mathbf{q}_1 & \mathbf{q}_n^T\mathbf{q}_2 & \cdots & \mathbf{q}_n^T\mathbf{q}_n \end{pmatrix}. \quad (6.5)$$

So orthonormality $\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}$ is equivalent to $Q^T Q = I_n$, where I_n is the $n \times n$ identity matrix. □

▸ Note that the columns of Q are a basis for range$(Q) = \{Q\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\}$.

Example → The set $S = \left\{\frac{1}{\sqrt{5}}(2,1)^T, \frac{1}{\sqrt{5}}(1,-2)^T\right\}$.
The two vectors in S are orthonormal, since

$$\frac{1}{\sqrt{5}}\begin{pmatrix}2 & 1\end{pmatrix}\frac{1}{\sqrt{5}}\begin{pmatrix}2\\1\end{pmatrix} = 1, \quad \frac{1}{\sqrt{5}}\begin{pmatrix}1 & -2\end{pmatrix}\frac{1}{\sqrt{5}}\begin{pmatrix}1\\-2\end{pmatrix} = 1, \quad \frac{1}{\sqrt{5}}\begin{pmatrix}2 & 1\end{pmatrix}\frac{1}{\sqrt{5}}\begin{pmatrix}1\\-2\end{pmatrix} = 0.$$

Therefore S forms a basis for $\mathbb{R}^2$. If $\boldsymbol{x}$ is a vector with components x_1, x_2 in the standard basis $\{(1,0)^T, (0,1)^T\}$, then the components of $\boldsymbol{x}$ in the basis given by S are

$$\begin{pmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

▸ Inner products are preserved under multiplication by orthogonal matrices, since $(Q\boldsymbol{x})^T Q\boldsymbol{y} = \boldsymbol{x}^T(Q^T Q)\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{y}$. This means that angles between vectors and the lengths of vectors are preserved. Multiplication by an orthogonal matrix corresponds to a rigid rotation (if $\det(Q) = 1$) or a reflection (if $\det(Q) = -1$).

Theorem 6.3. *The matrix $A^T A$ is invertible iff the columns of A are linearly independent, in which case $A\boldsymbol{x} = \boldsymbol{b}$ has a unique least-squares solution $\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$.*

Proof. If $A^T A$ is singular (non-invertible), then $A^T A\boldsymbol{x} = \boldsymbol{0}$ for some non-zero vector $\boldsymbol{x}$, implying that

$$\boldsymbol{x}^T A^T A\boldsymbol{x} = 0 \implies \|A\boldsymbol{x}\|_2^2 = 0 \implies A\boldsymbol{x} = \boldsymbol{0}. \quad (6.8)$$

This implies that A is rank-deficient (i.e. its columns are linearly dependent).

Conversely, if A is rank-deficient, then $A\boldsymbol{x} = \boldsymbol{0}$ for some $\boldsymbol{x} \neq 0$, implying $A^T A\boldsymbol{x} = \boldsymbol{0}$ and hence that $A^T A$ is singular. □

▸ The $n \times m$ matrix $(A^T A)^{-1} A^T$ is called the *Moore-Penrose pseudoinverse* of A. In practice, we would solve the normal equations (6.7) directly, rather than calculating the pseudoinverse itself.

QR Decomposition (6.3)

Least Square Approximation-2

27/04/2025 Sunday

$Ax=b$

$A^TAx = A^Tb$

$x = (A^TA)^{-1}A^Tb$

inverting is a hassle

$Ax=b$

$A=QR$

R = Upper triangular matrix

Q = All columns form an orthonormal set

$A=BQR$

$Q = \begin{bmatrix} 2 & 4 \\ 3 & 5 \\ 1 & 6 \end{bmatrix}$ $q_1 \quad q_2$

$\vec{q_1} = (2,3,1)^T$, $\vec{q_2} = (4,5,6)^T$

Theorem $Q^TQ = I$ → Identity matrix

2×2 $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

3×3 $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

In practice the normal matrix A^TA can often be ill conditioned. Thus we do QR decomposition.

Gram-Schmidt Process

Q) From these given two vectors, find the orthonormal vectors using Gram-Schmidt process.

given $A = \begin{bmatrix} 3 & 1 \\ 6 & 2 \\ 0 & 2 \end{bmatrix}$ or $u_1 = (3,6,0)^T$, $u_2 = (1,2,2)^T$

$$P_k = u_k - \sum_{i=1}^{k-1} (u_k^T \cdot q_i) q_i, \quad q_k = \frac{P_k}{|P_k|}$$

Ans $P_1 = u_1 = \begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix}$, $q_1 = \frac{P_1}{|P_1|} = \frac{\begin{pmatrix} 3 \\ 6 \\ 0 \end{pmatrix}}{\sqrt{3^2+6^2+0^2}}$

$P_2 = u_2 - (u_2^T q_1) q_1$, $q_2 = \frac{P_2}{|P_2|} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

$= \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} - \left\{ (1\ 2\ 2) \cdot \frac{1}{\sqrt{45}} \begin{pmatrix} 3 \\ 6 \\ 0 \end{pmatrix} \right\} \frac{1}{\sqrt{45}} \begin{pmatrix} 3 \\ 6 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$ $\therefore Q = [q_1\ q_2]$

$Ax = b$, $A = QR$

$\Rightarrow A^TAx = A^Tb$

$\Rightarrow (QR)^T QRx = (QR)^T b$

$\Rightarrow Q^TR^TQRx = R^TQ^Tb$

$\Rightarrow (Q^TQ)R^TRx = R^TQ^Tb$

$\Rightarrow Rx = Q^Tb$ ✓

$\therefore x = R^{-1}Q^Tb$

Q) Given $(-3,0)$, $(0,0)$, $(6,2)$ find $P_1(x)$

3 nodes but degree 1 : Overdetermined system

Ans: $\begin{pmatrix} 1 & -3 \\ 1 & 0 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$

A x b

$u_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $u_2 = \begin{pmatrix} -3 \\ 0 \\ 6 \end{pmatrix}$

$P_1 = u_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, $q_1 = \frac{P_1}{|P_1|} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix}$

$P_2 = u_2 - (u_2^T q_1) q_1$

$= \begin{pmatrix} -3 \\ 0 \\ 6 \end{pmatrix} - \left[(-3\ 0\ 6) \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix} \right] \begin{pmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{pmatrix} = \begin{pmatrix} -4 \\ -1 \\ 5 \end{pmatrix}$, $q_2 = \frac{P_2}{|P_2|} = \begin{pmatrix} -4/\sqrt{42} \\ -1/\sqrt{42} \\ 5/\sqrt{42} \end{pmatrix}$

$\therefore Q = \begin{pmatrix} 1/\sqrt{3} & -4/\sqrt{42} \\ 1/\sqrt{3} & -1/\sqrt{42} \\ 1/\sqrt{3} & 5/\sqrt{42} \end{pmatrix}$

$\vec{u} \cdot \vec{v} = 0$ → Orthogonal → Orthonormal set

As column of $Q = 2$, R will be 2×2 matrix

$\therefore R = 2\times2$ upper triangular matrix

$R = \begin{pmatrix} u_1^Tq_1 & u_2^Tq_1 \\ 0 & u_2^Tq_2 \end{pmatrix} = \begin{pmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{42} \end{pmatrix}$

OR

☆ $A = QR$

$Q^TA = Q^TQR$

$\therefore Q^TA = R$ ✓

easier

$\therefore R = \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ -4/\sqrt{42} & -1/\sqrt{42} & 5/\sqrt{42} \end{pmatrix} \begin{pmatrix} 1 & -3 \\ 1 & 0 \\ 1 & 6 \end{pmatrix}$ (Q^T, A)

$= \begin{pmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{42} \end{pmatrix}$ same results

$Rx = Q^Tb$

$\begin{pmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{42} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ -4/\sqrt{42} & -1/\sqrt{42} & 5/\sqrt{42} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$

$\Rightarrow \begin{pmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{42} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{3} \\ 10/\sqrt{42} \end{pmatrix}$

$\therefore \sqrt{42}\, a_1 = 10/\sqrt{42}$ | $\sqrt{3}\, a_0 + \sqrt{3}\, a_1 = 2/\sqrt{3}$

$a_1 = 5/21$ | $a_0 = 3/7$

$\therefore P_1(x) = a_0 + a_1 x$

$= 3/7 + 5/21\, x$

Numerical Integration-1

01/05/2025 Thursday

$I(f) = \int_a^b f(x)\,dx$, $I_n(f) = \int_a^b P_n(x)\,dx$ (To computer, it's polynomial)

$P_n(x) = \sum_{k=0}^{n} l_k(x) f(x_k)$ (Lagrange basis)

$\therefore I_n(f) = \int_a^b \sum_{k=0}^{n} l_k(x) f(x_k)\,dx = \sum_{k=0}^{n} f(x_k) \int_a^b l_k(x)\,dx$ ($f(x_k)$: constant)

weight factor, $\delta_k = \int_a^b l_k(x)\,dx$

$I_n(f) = \sum_{k=0}^{n} \delta_k f(x_k)$ — Newton-Cotes formula (7.1)

*** The nodes given, will all be equidistant to each other. $a = x_0 < x_1 < x_2 \dots < x_n = b$

The formula for these kind of nodes is called Closed Newton-Cotes Formula. $h = \frac{b-a}{n}$

When $a < x_0 < x_1 < x_2 < \dots x_n < b$, it's called Open Newton-Cotes Formula. $h = \frac{b-a}{n+2}$

When in Newton-Cotes formula $n=1$ (1 degree polynomial) $P_1(x)$ then $h = b-a$. This is called Trapezium Rule. $x_0 = a$, $x_1 = b$ (Closed Newton-Cotes formula with $n=1$)

$P_1(x) = L_0(x) \cdot f(x_0) + L_1(x) + f(x_1)$
$= \frac{x-x_1}{x_0-x_1} f(a) + \frac{x-x_0}{x_1-x_0} f(b)$
$= \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$

$\delta_0 = \int_a^b L_0(x_0)\,dx = \int_a^b \frac{x-b}{a-b}\,dx = \frac{1}{a-b}\int_a^b x-b\,dx = \frac{1}{a-b}\left[\frac{x^2}{2} - bx\right]_a^b = \frac{1}{a-b}\left(\frac{b^2}{2} - b^2 - \frac{a^2}{2} - ab\right) = \frac{b-a}{2}$

$\delta_1 = \int_a^b L_1(x_0)\,dx = \int_a^b \frac{x-a}{b-a} = \frac{b-a}{2}$

$I_n(f) = \sum_{k=0}^{n} \delta_k f(x_k)$

$I_n(f) = \delta_0 f(x_0) + \delta_1 f(x_1) = \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$

$\therefore I_n(f) = \frac{b-a}{2}\left[f(a) + f(b)\right]$

Q) $a=0$, $b=2$, $f(x) = e^x$

Ans $I = \int_a^b f(x)\,dx = \int_0^2 e^x\,dx = \left[e^x\right]_0^2 = e^2 - e^0 = 6.389$

$I_1 = \frac{b-a}{2}\left[f(a) + f(b)\right] = \frac{2-0}{2}\left[e^0 + e^1\right] = 8.389$

Relative error $= \frac{|I - I_1|}{|I|} \times 100 = \frac{6.389 - 8.389}{6.389} \times 100 = 31.3\%$

☆ As 1 degree Polynomial, thus high relative error. Higher degree will have higher Accuracy

Upper Bound $= \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)$, $\xi \in [a,b]$

(for integration) $|I - I_n| \le \left|\frac{f^{(n+1)}(\xi)}{(n+1)!}\right|_{\max \xi \in [a,b]} \times \int_a^b (x-x_0)(x-x_1)\cdots(x-x_n)\,dx$

Q) $a=0$, $b=2$, $f(x) = e^x$, $n=1$

Ans $\left|\frac{f^{(n+1)}(\xi)}{(n+1)!}\right|_{\max} \to \left|\frac{f^{(2)}(\xi)}{2!}\right|_{\max} = \frac{1}{2} e^{\xi}\Big|_{\max \xi \in [0,2]} = \frac{e^2}{2}$ (max when $x=2$)

$\int_0^2 (x-0)(x-2)\,dx = \int_0^2 (x^2 - 2x)\,dx = \left[x^3/3 - 2x^2/2\right]_0^2 = \frac{4}{3}$

$\therefore$ Upper bound $\le e^2/2 \times 4/3 = 4.926$ //

▸ Theorem 7.1 suggests that the accuracy of I_n is limited both by the smoothness of f (outside our control) and by the location of the nodes x_k. If the nodes are free to be chosen, then we can use Gaussian integration (see later).

▸ As with interpolation, taking a high n is not usually a good idea. One can prove for the closed Newton-Cotes formula that

$$\sum_{k=0}^{n} |\sigma_k| \to \infty \quad \text{as} \quad n \to \infty.$$

This makes the quadrature vulnerable to rounding errors for large n.

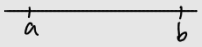
Composite Newton-Cotes Formula (7.2) Numerical Integration-2

$[a,b]$ *** Here make some extra points bet^n a&b to make the accuracy better.

$n=1$

2 nodes

$m \uparrow$ Accuracy $\uparrow$



m = no. of times ib is divided → $m=1$ means no division

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

$m+1$ nodes

☆ Only 1 degree polynomials

Q) $a=0, b=2$, $f(x)=e^x$, $m=2$. Use composite newton-cotes Formula

$I = \int_0^2 e^x dx = 6.389$ $\quad h = \frac{2-0}{2} = 1$, $x_0=0$, $x_1 = 0+1=1$, $x_2 = 1+1=2$

$C_{1,2} = h/2\left[f(x_0)+2f(x_1)+f(x_2)\right]$

$= h/2\left[e^0+2e^1+e^2\right] = 6.9126$

h halves → error decreases by 4

$\therefore O(h^2)$ Quadratic Convergence ☆

Q) $m=3$, $h = \frac{2-0}{3} = \frac{2}{3}$

$x_0=0$, $x_1 = 0+\frac{2}{3} = \frac{2}{3}$, $x_2 = \frac{2}{3}+\frac{2}{3} = \frac{4}{3}$, $x_3 = \frac{4}{3}+\frac{2}{3} = 2$

$\therefore C_{1,3} = h/2\left[f(x_0)+2f(x_1)+2f(x_2)+f(x_3)\right]$

$= \frac{2}{3/2}\left[e^0+2e^{2/3}+2e^{4/3}+e^2\right] = 6.623$

Q) $m=4$, $h = \frac{2-0}{4} = \frac{1}{2}$

$x_0=0$, $x_1 = 0+\frac{1}{2} = \frac{1}{2}$, $x_2 = 1$, $x_3 = \frac{3}{2}$, $x_4 = 2$

$\therefore C_{1,4} = \frac{0.5}{2}\left[f(x_0)+2\left(f(x_1)+f(x_2)+f(x_3)\right)+f(x_4)\right] = 6.5216$

closer to the actual value

$C_{1,m} = h/2\left[f(x_0)+2f(x_1)+\dots+2f(x_{m-1})+f(x_m)\right]$

1 degree

$n=1$ → Trapezoidal Rule

$n>1$ → Simpson's Rule

*** Using Simpson's Rule, b/ll $n=2$ & 3 we can get the exact value.

Simpson's Rule (Derivation is Important)

(This is the $n=2$ Newton-Cotes formula)

Simpson's rule

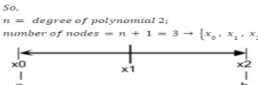
From, Trapezium / Trapezoidal rule → $I_1(f) = \int_a^b P_1(x)\,dx$

Simpson's rule → $I_2(f) = \int_a^b P_2(x)\,dx$

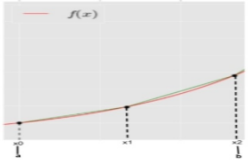
So,

n = degree of polynomial 2;

number of nodes = $n+1 = 3 \rightarrow \{x_0, x_1, x_2\}$



$m = x_1 = x_0 + h = \frac{a+b}{2}$



$I_2(f) = \int_a^b P_2(x)\,dx$

$I_2(f) = \int_a^b \left[l_0(x)\,f(x_0) + l_1(x)\,f(x_1) + l_2(x)\,f(x_2)\right] dx$

$\Rightarrow I_2(f) = \int_a^b l_0(x)\,dx\,f(x_0) + l_1(x)\,dx\,f(x_1) + l_2(x)\,dx\,f(x_2)$

$\Rightarrow I_2(f) = \sigma_0 f(x_0) + \sigma_1 f(x_1) + \sigma_2 f(x_2)$

$\Rightarrow I_2(f) = \sigma_0 f(a) + \sigma_1 f(m) + \sigma_2 f(b)$

Here,

$\sigma_0 = \int_a^b l_0(x)\,dx$

$\Rightarrow \sigma_0 = \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\,dx$

$\Rightarrow \sigma_0 = \int_a^b \frac{(x-m)(x-b)}{(a-m)(a-b)}\,dx$

$\Rightarrow \sigma_0 = \frac{1}{(a-m)(a-b)}\int_a^b (x-m)(x-b)\,dx$

$\Rightarrow \sigma_0 = \frac{1}{(a-m)(a-b)} \times \frac{(b-a)(a-b)(a-m)}{6}$

$\Rightarrow \boxed{\sigma_0 = \frac{b-a}{6}}$

$\sigma_1 = \int_a^b l_1(x)\,dx$

$\Rightarrow \sigma_1 = \int_a^b \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\,dx$

$\Rightarrow \sigma_1 = \int_a^b \frac{(x-a)(x-b)}{(m-a)(m-b)}\,dx$

$\Rightarrow \sigma_1 = \frac{1}{(m-a)(m-b)}\int_a^b (x-a)(x-b)\,dx$

$\Rightarrow \sigma_1 = \frac{1}{(m-a)(m-b)} \times \frac{2(b-a)(m-a)(m-b)}{3}$

$\Rightarrow \boxed{\sigma_1 = \frac{4}{6}(b-a)}$

$\sigma_2 = \int_a^b l_2(x)\,dx$

$\Rightarrow \sigma_2 = \int_a^b \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\,dx$

$\Rightarrow \sigma_2 = \int_a^b \frac{(x-a)(x-m)}{(b-a)(b-m)}\,dx$

$\Rightarrow \sigma_2 = \frac{1}{(b-a)(b-m)}\int_a^b (x-a)(x-m)\,dx$

$\Rightarrow \sigma_2 = \frac{1}{(b-a)(b-m)} \times \frac{(b-a)(b-a)(b-m)}{6}$

$\Rightarrow \boxed{\sigma_2 = \frac{b-a}{6}}$

So,

$\Rightarrow I_2(f) = \sigma_0 f(x_0) + \sigma_1 f(x_1) + \sigma_2 f(x_2)$

$\Rightarrow I_2(f) = \sigma_0 f(a) + \sigma_1 f(m) + \sigma_2 f(b)$

$\Rightarrow I_2(f) = \frac{b-a}{6} f(a) + \frac{2(b-a)}{3} f\left(\frac{a+b}{2}\right) + \frac{b-a}{6} f(b)$

$\Rightarrow I_2(f) = \frac{b-a}{6} f(a) + \frac{4(b-a)}{6} f\left(\frac{a+b}{2}\right) + \frac{b-a}{6} f(b)$

$\boxed{I_2(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]}$ → Simpson's formula

1) Example:

Find $I(f)$ and $I_2(f)$ of the function e^x on interval $[0, 2]$. Show relative error.

Solution →

$I(f) = \int_0^2 e^x dx = [e^x]_0^2 = e^2 - e^0 = 6.389$ [Actual Integration]

$I_2(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$ [Approximate Integration]

$\Rightarrow I_2(f) = \frac{2-0}{6}\left[e^0 + 4e^{\frac{0+2}{2}} + e^2\right]$

$\Rightarrow I_2(f) = \frac{1}{3}\left[e^0 + 4e^1 + e^2\right] = 6.421$

relative error (in percentage) $= \frac{|I_2 - I|}{|I|} \times 100 = \frac{|6.421-6.389|}{|6.389|} \times 100 = 0.501\%$

2) Example:

Compute the upper bound error for I_2 of the function e^x on interval $[0, 2]$.

Solution →

Here, $n = 2$

$f(x) = e^x$

$[a, b] = [0, 2]$

For integration, upper bound error →

$|I - I_n| \le \left|\frac{f^{(n+1)}\xi}{(n+1)!}\right| \int_a^b |(x-x_0)(x-x_1)\dots\dots(x-x_n)|\,dx$

$\Rightarrow |I - I_2| \le \left|\frac{f^{(3)}\xi}{3!}\right| \int_0^2 |(x-x_0)(x-x_1)(x-x_2)|\,dx$

$\Rightarrow |I - I_2| \le \left|\frac{f^{(3)}\xi}{3!}\right| \int_0^2 |(x-a)(x-\frac{a+b}{2})(x-b)|\,dx$

Now, finding the max of $\left|\frac{f^{(3)}\xi}{3!}\right|$ within $[0, 2]$

$= \left|\frac{f^3\xi}{3!}\right| = \left|\frac{e^\xi}{6}\right| = \left|\frac{e^\xi}{6}\right| = \frac{e^\xi}{6}$ [where $|f^3(x)| = |e^x|$]

Now, finding the max of $\int_0^2 |(x-a)(x-\frac{a+b}{2})(x-b)|\,dx$ within $[0, 2]$

$= \int_0^2 |(x-a)(x-\frac{a+b}{2})(x-b)|\,dx$

$= \int_0^2 |(x-0)(x-\frac{0+2}{2})(x-2)|\,dx$

$= \int_0^2 |(x)(x-1)(x-2)|\,dx$

$= \int_0^2 |(x^3 - 3x^2 + 2x)|\,dx$

$= \left|\left[\frac{x^4}{4} - \frac{3x^3}{3} + \frac{2x^2}{2}\right]_0^2\right| = 2$

So, upper bound of error $\le \frac{e^\xi}{6} \times 2 \approx 2.463$


THE
END