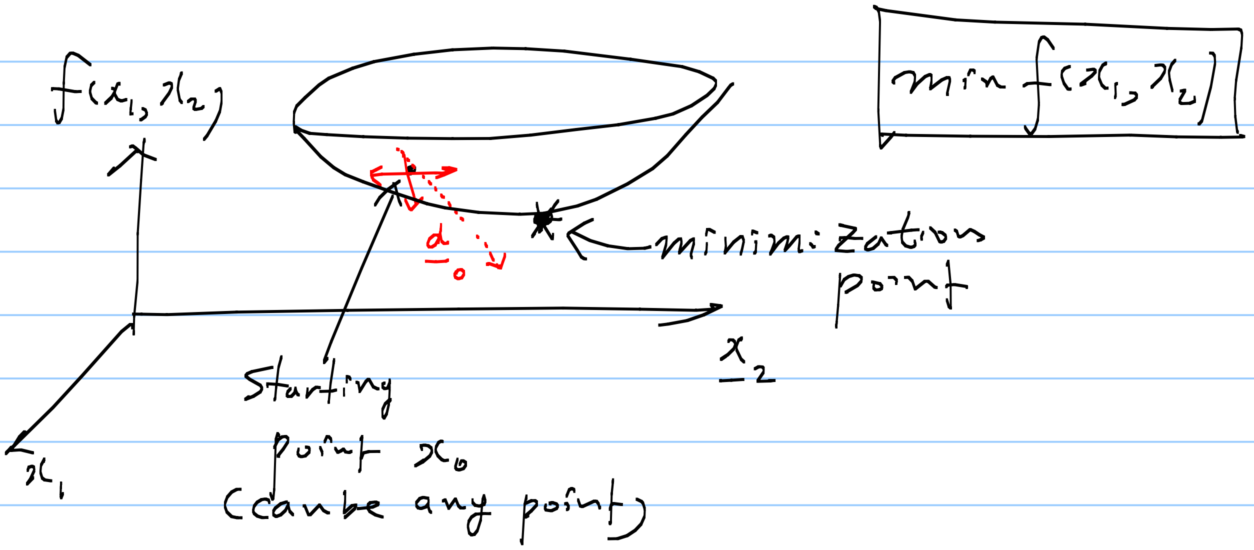
2-2. Numerical Method for 1-D Constrained problem

Motivation to study 1-D Constrained problems ?



Typical Search Method

① Look for the direction of the steepest descent

② find α (scalar) to minimize $f(\underset{\sim}{x}_0 + \alpha \underset{\sim}{d}_0)$

$\Rightarrow$ Step ② is one-dimensional problem (i.e.; the min problem involves only one design variable α) $\rightarrow \alpha_0^*$

③ update the search point

$$\underset{\sim}{x}_{New} = \underline{x}_{OLD} + \alpha_0^* \underline{d}_0$$

$\Uparrow$

$\underline{x}_0$ for 1st iteration

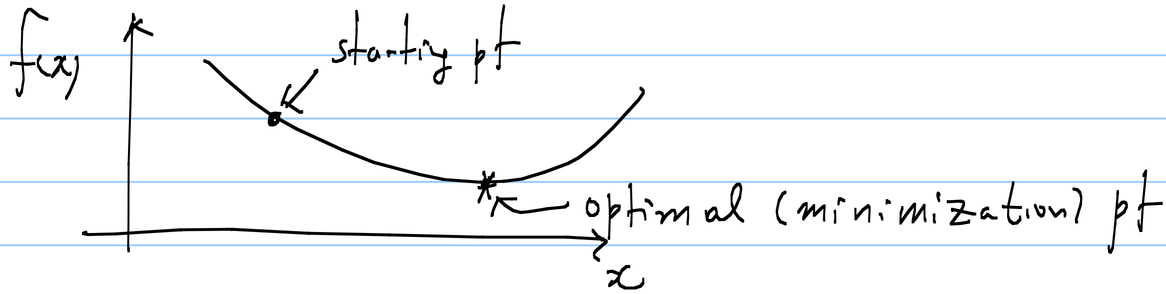
④ repeat until convergence

Remark: Two key components in optimization algorithm

① Search direction $\underline{d}$

② 1-D minimization

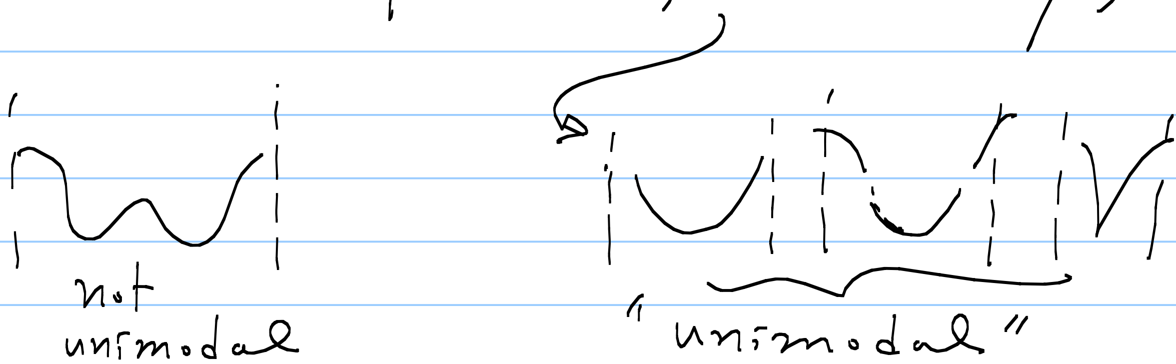
※ We will begin with 1-D Numerical minimization



We will study

Phase 1: Bracketing

Phase 2: Minimization for a given internation where there is only one minimization point (i.e, "unimodality")



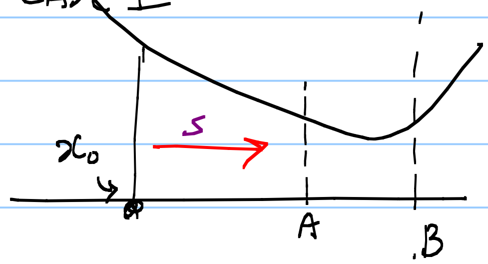
For Phase 2, will study:

1. elimination method such as Golden Section Method
2. polynomial Interpolation method (quadratic or Cubic polynomial approx.)

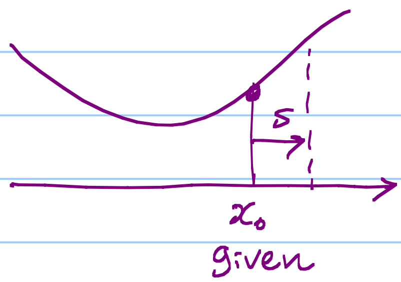
Phase 1: Bracketing

For given x_0 (initial guess) and s (search interval), Determine $I = [A, B]$ to apply 1-D search Algorithm

CASE I



CASE II



<Bracketing strategy>

- ① $f(x_1) = f(x_0 + s)$

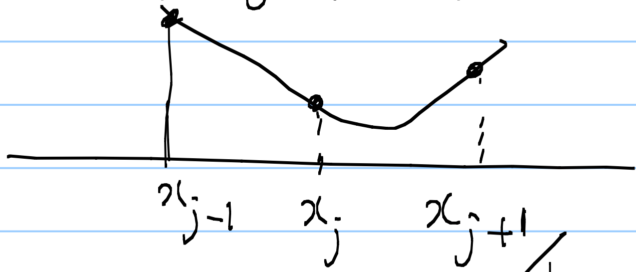
  if $f(x_1) > f(x_0) \rightarrow$ case II

  then $s = -s$

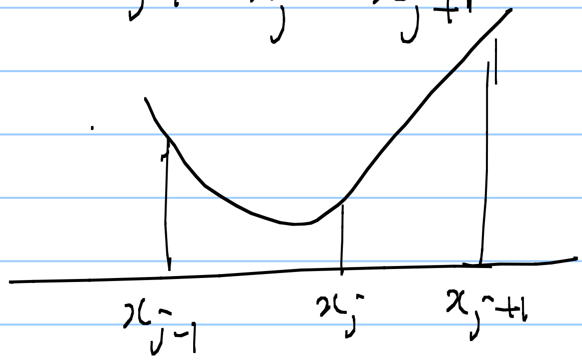
2. evaluate

$f(x)$ at $x_j = x_{j-1} + s$

until $f(x_{j+1}) > f(x_j)$



(or



)

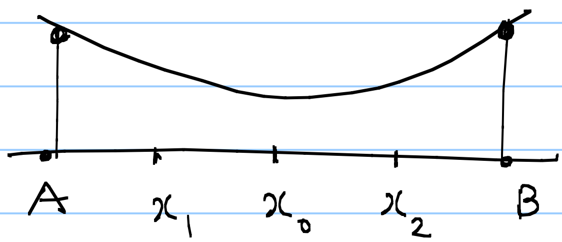
3. Then $I = [A, B] \equiv [x_{j-1}, x_{j+1}]$

Phase 2 : Minimization by "Elimination"

☒ Will study → { Interval halving Method
Golden Section Method

for a given interval $I = [A, B]$

2-1 Interval halving Method



① Choose

$$x_0 = \frac{1}{2}(A+B)$$

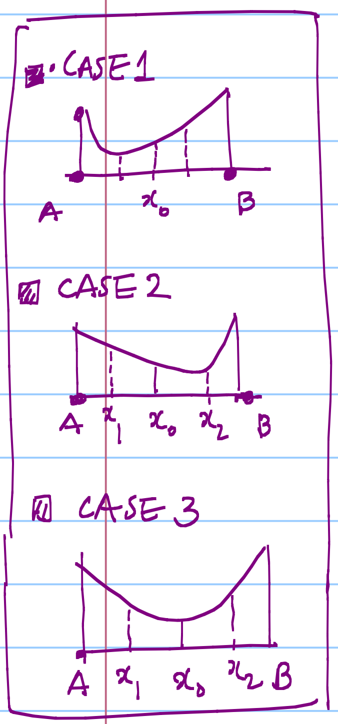
$$x_1 = A + \frac{1}{4}(B-A)$$

$$x_2 = B - \frac{1}{4}(B-A)$$

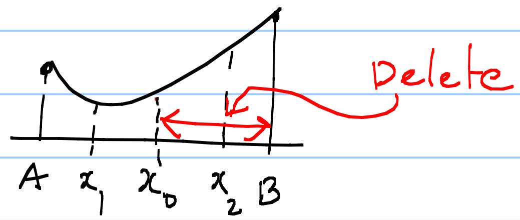
Compute $f_0 = f(x_0)$, $f_1 = f(x_1)$

and $f_2 = f(x_2)$

② Then check three possible cases and reduce the search interval I



Case 1 : if



i.e.

$$\text{if } f_2 > f_0 > f_1,$$

- Delete $[x_0, B]$

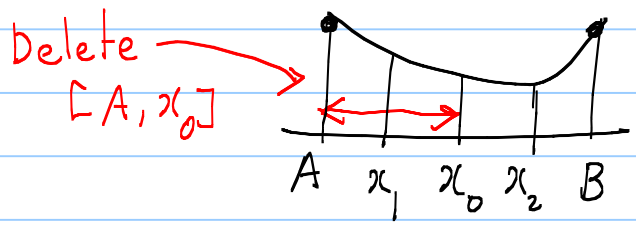
  $I_1 = [A, x_0] \equiv [A, B]$

- set : $x_0 \leftarrow x_1$

  $f_0 \leftarrow f_1$

  (To save computation time)

CASE 2 : elseif



i.e.: if $f_2 < f_1 < f_0$,

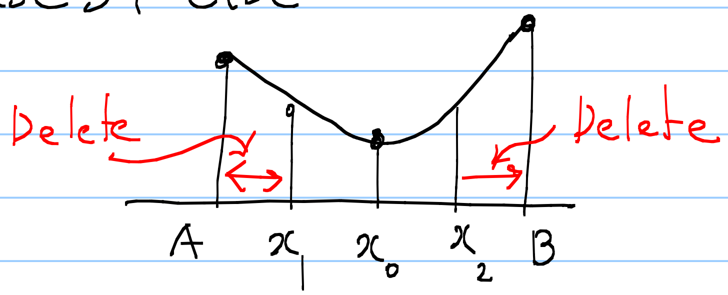
- Delete $[A, x_0]$

  $I_1 = [x_0, B] \equiv [A, B]$

  Set: $x_0 \leftarrow x_1$

  $f_0 \leftarrow f_1$

Case 3: else



else (i.e, $f_1 > f_0$ and $f_2 > f_0$)

- Delete $[A, x_1]$ and $[x_2, B]$
- $I_1 = [x_1, x_2] = [A, B]$

(Set $x_0 \Leftarrow x_0$

$f_0 \Leftarrow f_0$)

③ Convergence Check?

if $|I_l| \le \epsilon$, Converged

given small parameter

else, repeat

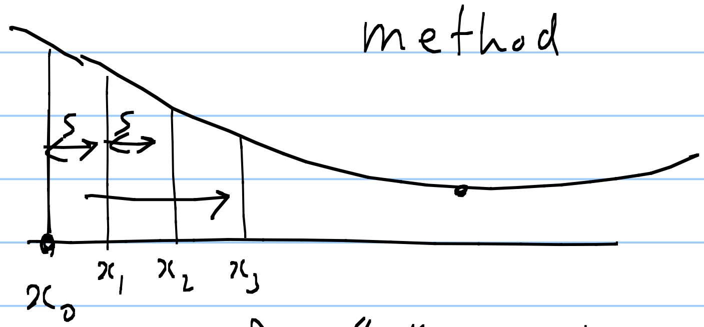
Observation:

- Except the first iteration step, two functions calculations are needed at every iteration.
- Thus, one may relate the Interval size and function evaluation number n as

k (iter. No)	n (Total ftn Eval. No)	Interval
1	3	$I_0 \times \frac{1}{2}$
2	5	$I_0 \times \left(\frac{1}{2}\right)^2$
3	7	$I_0 \times \left(\frac{1}{2}\right)^3$

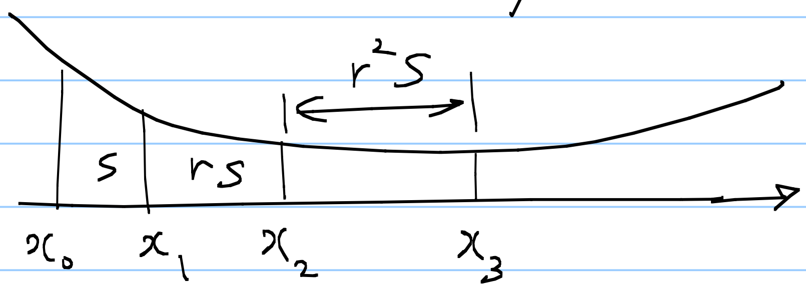
$$\boxed{I(n) \longrightarrow I_0 \times \left(\frac{1}{2}\right)^{\frac{n-1}{2}}}$$

Remark on: Uniform Bracketing and Internal halving method



→ uniform "S"; can be inefficient to bracket the minimum point

⇒ ① Increase search interval at every step for speedup although the initial search interval I has large uncertainty



$r > 1$

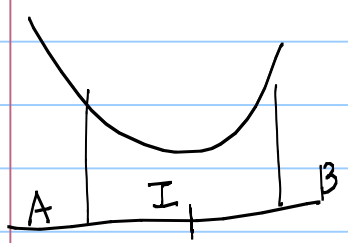
② Commonly choose r = Golden section ratio ≈ 1.618 (will see later) an elimination method

$\boxed{2-2}$ Golden Section Method

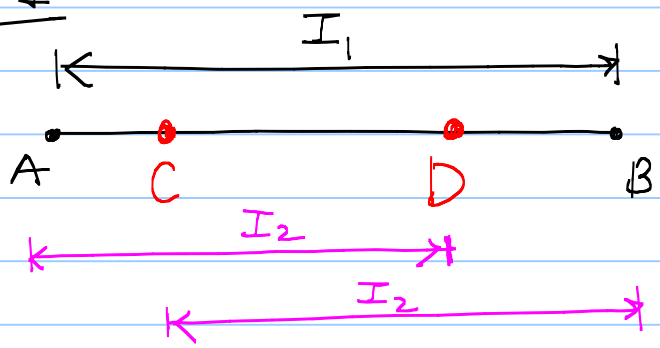
① a robust 1-D method

② a zeroth-order method
(no derivative of $f(x)$ is required)

[cf: 1st-order method]



□ Step 1



Choose 2 new points $C, D \in I_1$ such that

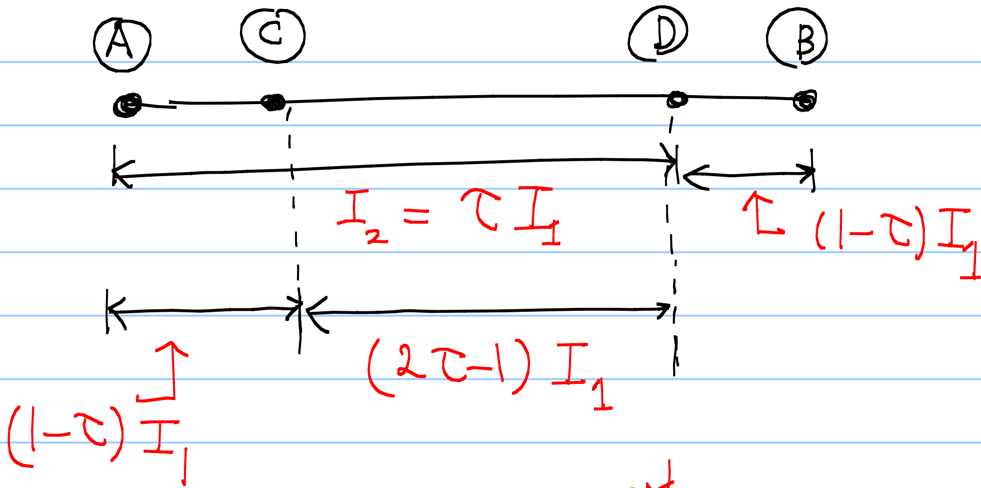
- $\overline{AD} = \overline{BC} = I_2$
- Thus $I_2 = \tau I_1$

yet unknown but $0 < \tau < 1$

• Next search interval

$I_2 = [A, D]$ or $[C, B]$

* observation



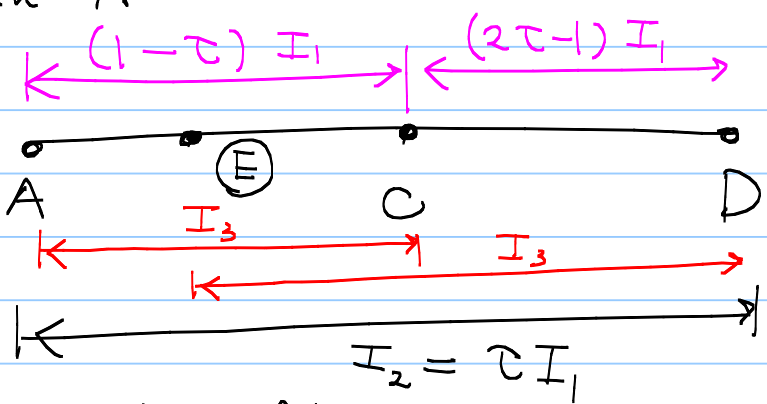
only one

** ☐ Step 2: Choose "One" New point E to reduce I_2 to I_3

* Assume $I_2 = [A, D]$

* Two possible locations of E

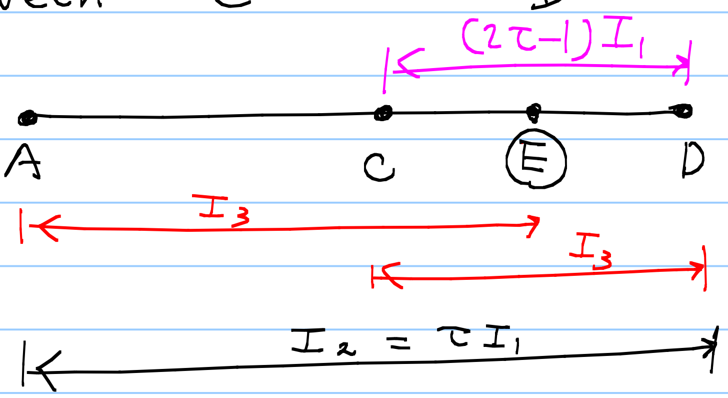
CASE ①: Between A and C



For Case ① to Be Meaningful,

$$(1-\tau)I_1 > (2\tau-1)I_1 \Rightarrow \boxed{\tau < \frac{2}{3}} \quad \text{--- (a)}$$

CASE ②: Between C and D



For CASE ② to be meaningful, $\boxed{\tau > \frac{2}{3}}$

Analysis : case 1

i) From Figure,

$$I_3 = (1-\tau) I_1 \quad \text{------} \quad (a)$$

ii) But we require

$$I_3 = \tau I_2 = \tau \cdot (\tau I_1)$$
$$= \tau^2 I_1 \quad \text{-----} \quad (b)$$

(a) = (b)

$$\therefore (1-\tau) I_1 = \tau^2 I_1$$
$$\tau^2 + \tau - 1 = 0$$
$$\tau = (\sqrt{5}-1)/2$$
$$= 0.61803 < \frac{2}{3} \text{ ; okay}$$

$$\left(\tau = \frac{1}{1+\tau}\right) \Uparrow$$

"Golden section Ratio"

Analysis : case 2

i) From Figure

$$I_3 = (2\tau-1) I_1 \quad \text{----} \quad (a)$$

$$I_3 = \overline{AE} = \overline{CD}$$

ii) We require

$$I_3 = \tau I_2 = \tau^2 I_1 \quad \text{--} \quad (b)$$

(a) = (b)

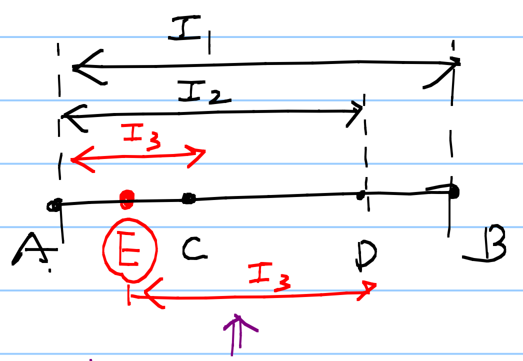
$$\tau^2 = 2\tau - 1 \rightarrow (\tau-1)^2 = 0$$

$$\tau = 1 \rightarrow \text{not meaningful}$$

$$(0 < \tau < 1)$$

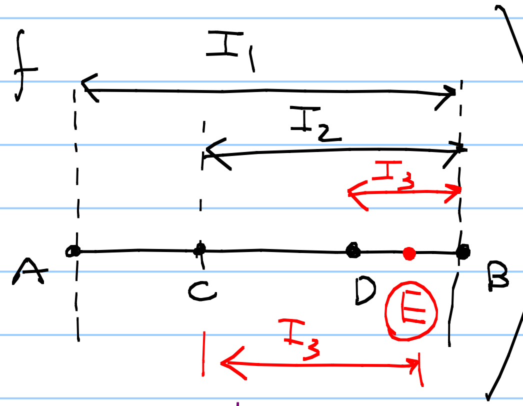
Observation

if



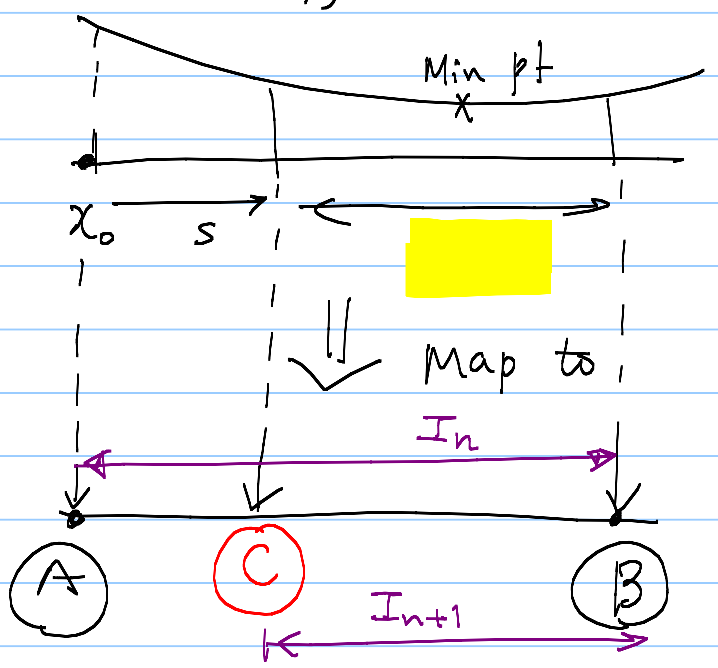
E lies between A and C.

if



E lies between D and B.

$\boxed{4}$ To combine bracketing and Golden section method efficiently, increase search step as



① If $f_B > f_C$, we can start the golden section method

$\Rightarrow$ Choose r such that

$$\frac{I_{n+1}}{I_n} = \tau \equiv \frac{rs}{(1+r)s}$$

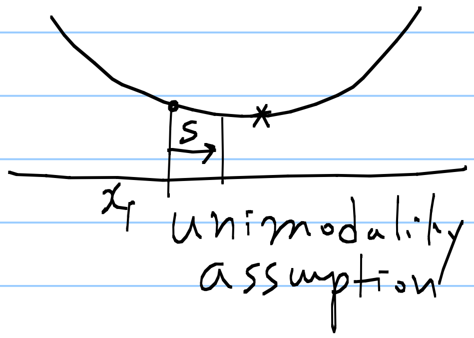
$\rightarrow$ solving: $\boxed{r = \frac{1}{\tau} = 1.618034\cdots}$

(17)

Integrated Bracketing and Golden-Section Algorithm

Basic (Simple) Search method

- Bracketing, then golden sectioning



[Algorithm]

■ Step 1 : user gives

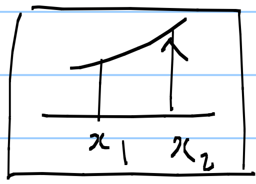
$\begin{cases} x_1 \text{ (starting point)} \\ s \text{ (search interval)} > 0 \end{cases}$

☑ Step 2 : function evaluations

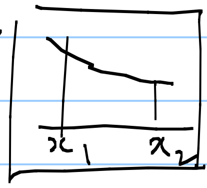
$$f_1 = f(x_1)$$

$$x_2 = x_1 + s \; ; \; f_2 = f(x_2)$$

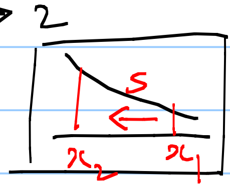
☑ Step 3 : check search direction

- if $f_2 \geq f_1 \Leftrightarrow$ 

  go to step 4

- else ($f_2 < f_1$) $\Leftrightarrow$ 

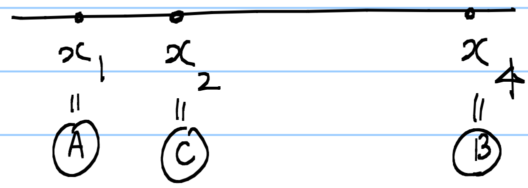
  - $s = -s$
  - Interchange $1 \leftrightarrow 2$

  $\Rightarrow$ 

step 4 : Guess the end point of the search Interval

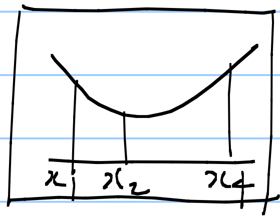
$$s = rs = \frac{s}{\tau} \qquad \text{note } (\tau < 1)$$

$$x_4 = x_2 + s$$



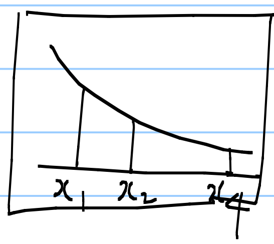
step 5 : Check function behavior

- if $f_4 > f_2 \Leftrightarrow$



Go To step 7
(i.e, Start golden section search)

- else ($f_4 < f_2$) $\Leftrightarrow$



  - Keep bracketing

  Go To STEP 6

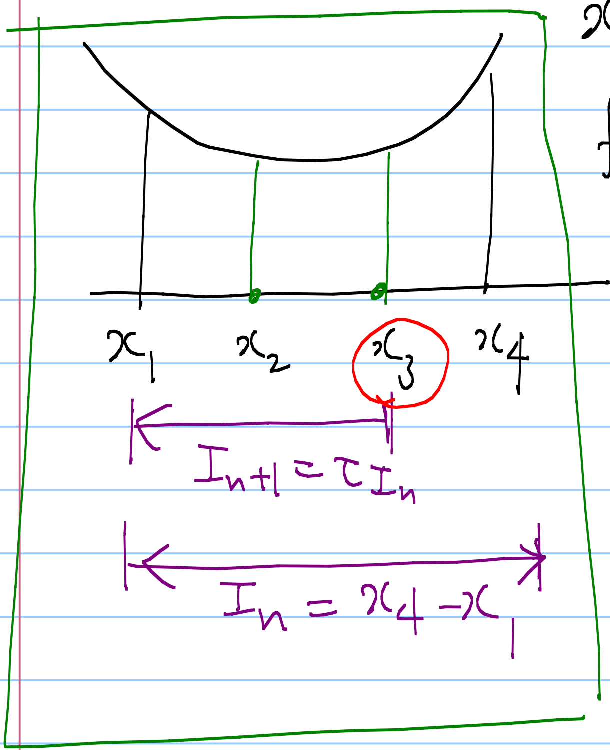
☐ STEP 6 : Reset Variables to Continue Bracketing

$$x_1 \Leftarrow x_2 , \quad f_1 \Leftarrow f_2$$
$$x_2 \Leftarrow x_4 , \quad f_2 \Leftarrow f_4$$

and Go To step 4

☐ step 7 : Start Golden Section search

$$x_3 \overset{?}{=} (1-\tau) x_1 + \tau x_4 \quad (*)$$
$$f_3 = f(x_3)$$

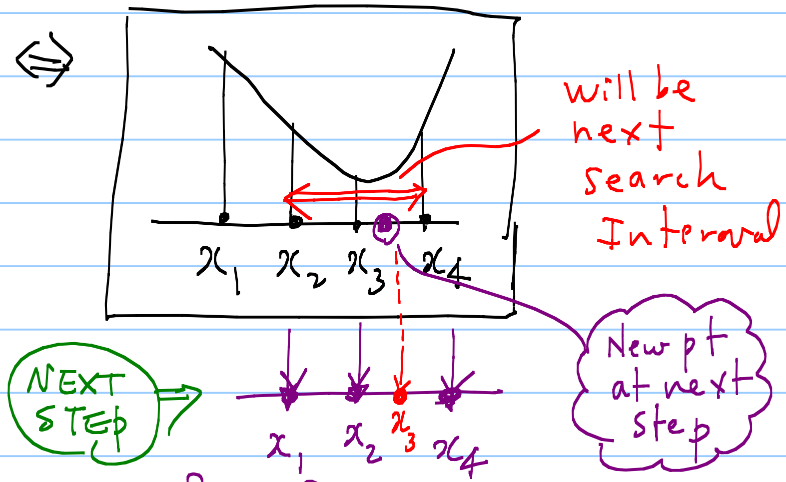


why $(*) = ?$

$$\begin{aligned} x_3 &= x_1 + I_{n+1} \\ &= x_1 + \tau I_n \\ &= x_1 + \tau (x_4 - x_1) \\ &= (1-\tau) x_1 + \tau x_4 \end{aligned}$$

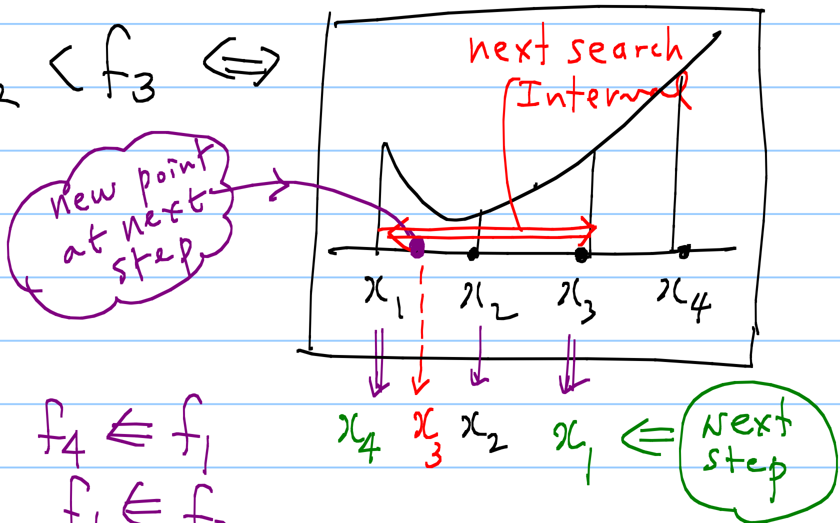
[1] Step 8 : Reduce Search Interval

if $f_2 > f_3$ $\Leftrightarrow$



- Set $x_1 \Leftarrow x_2$, $f_1 \Leftarrow f_2$
  $x_2 \Leftarrow x_3$, $f_2 \Leftarrow f_3$
- Go to 9 for convergence check

else $f_2 < f_3$ $\Leftrightarrow$



- set $x_4 \Leftarrow x_1$, $f_4 \Leftarrow f_1$
  $x_1 \Leftarrow x_3$, $f_1 \Leftarrow f_3$
- Go To 9

☑ Step 9 : Convergence check

satisfied : END

if not : go to step 7

(Golden section again)

Convergence criteria

usually check once → ① Interval size

$$|x_1 - x_3| \le \epsilon^x_{rel} |x_2| + \epsilon^x_{ABS}$$

(Typically ; $\epsilon^x_{rel} = 10^{-16}$, $\epsilon^x_{ABS} = 10^{-4}$)

or (and)

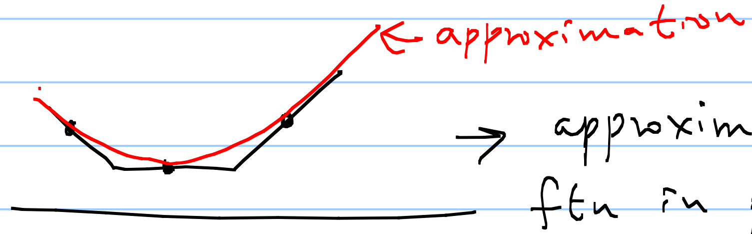
usually check over 2 successive iterations → ② function reduction

$$\bar{f} = \underline{f_1} + \underline{f_2} + \underline{f_3}$$

$$(\bar{f} - \bar{f}_{old}) \le \epsilon^f_{rel} |f_2| + \epsilon^f_{ABS}$$

(Typically; $\epsilon^f_{rel} = 10^{-16}$, $\epsilon^f_{ABS} = 10^{-6}$)

☐ Instead of Golden section Method, one may use polynominal-Based 1-D minimization method



→ approximate the ftn in polynominal
⇒ analytically determine min point.

- fast near minimum points
- 3-point quadratic approximation is popular, (if derivatives are also known, cubic polynomials can be used).

※ "Robust Quadratic Fit-Sectioning Algorithm" (by Brent)
→ highly recommeded
→ idea: use quadratic polynominal approx whenever possible; otherwise use golden section method.