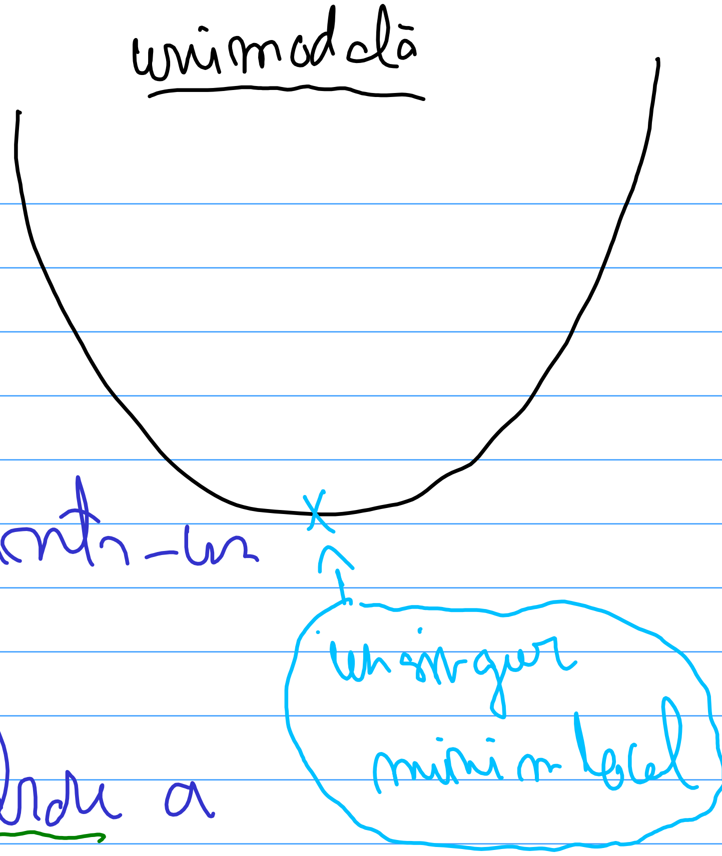
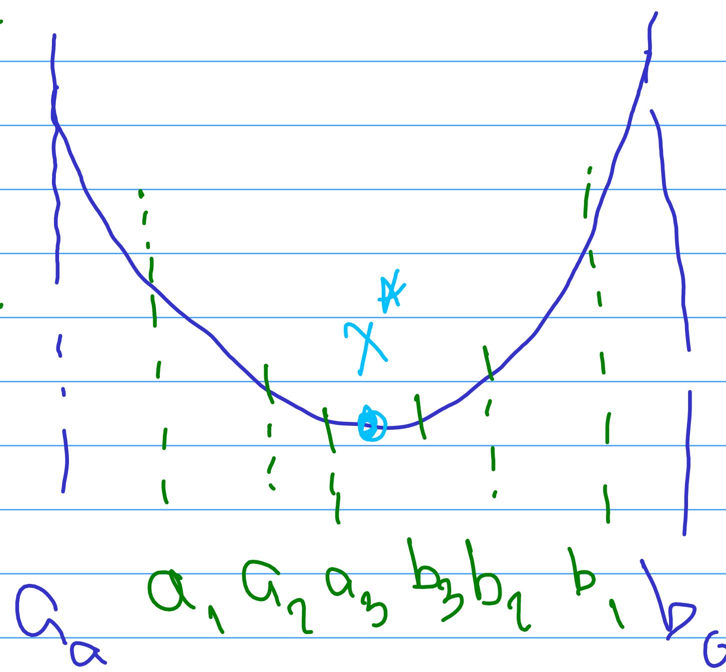
Alg. opt 1D

- un algoritm produce o serie de aproximații ale minimului printr-un proces iterativ.



- Algoritmi de încadrare a minimului: produc intervale $[a_n, b_n]$ care conțin soluția.



- f unimodală
- $[a,b]$ conține x^*
- avem $a < x_- < x_+ < b$

Dacă $f(x_-) \geq f(x_+) \Rightarrow x^* \in [x_-, b]$

Dacă $f(x_-) \leq f(x_+) \Rightarrow x^* \in [a, x_i^+]$

- Intuitiv: - algoritmul converge
  - la fiecare iterație eroarea scade.

Alg. optimizare
- convergență: eroarea tinde la 0
- viteza de convergență?

Clase de viteză de conv

$r_i = |x_i - x^*|$ (dist. de la punctul actual la soluție)

Liniară: $\frac{r_{i+1}}{r_i} \leq q < 1$ (raport de conv)

Subliniară: Dacă $r_i \to 0$ dar $\frac{r_{i+1}}{r_i}$ poate să se apropie de 1

Dacă: $r_i = \frac{1}{i} \Rightarrow \frac{r_{i+1}}{r_i} = \frac{i}{i+1} \to 1$

<u>Super liniară</u>: $\frac{\varepsilon_{i+1}}{\varepsilon_i} \xrightarrow[i \to \infty]{} 0$

<u>Conv pătratică</u>: $\varepsilon_{i+1} \approx C \varepsilon_i^2$

$$\varepsilon_i = 10^{-2}$$
$$\varepsilon_{i+1} = 10^{-4}$$
$$\varepsilon_{i+2} = 10^{-8}$$
$$\vdots$$

<u>Exemple</u>: $\gamma \in (0,1)$

- $\varepsilon_n = \gamma^n \to 0$ conv liniară

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} = \gamma < 1$$

- $\varepsilon_n = \gamma^{n^2} \longrightarrow 0$ conv superliniară

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} = \frac{\gamma^{(n+1)^2}}{\gamma^{n^2}} = \underset{0<\gamma<1}{\gamma^{2n+1}} \longrightarrow 0$$

$$r_{n+1} = \gamma^{(n+1)^2}$$
$$r_n^2 = \left(\gamma^{n^2}\right)^2 = \gamma^{2n^2} \qquad r_{n+1} \not\leq C r_n^2$$

- $r_n = \gamma^{2^n}$ conv. <u>pătratică</u>

$$r_{n+1} = \gamma^{2^{n+1}} = \left(\gamma^{2^n}\right)^2 = r_n^2$$

<u>Identificare grafică a ordinului de conv</u>

$r_i \rightsquigarrow$ funcție putere
dificil să găsim ordinul
din graficul lui r_i

<u>Ordinul</u> $= p$ a.î. $r_{i+1} \approx C r_i^p$

- reprezentăm r_{i+1} ca și funcție de r_i
- utilizăm o scară logaritmică.

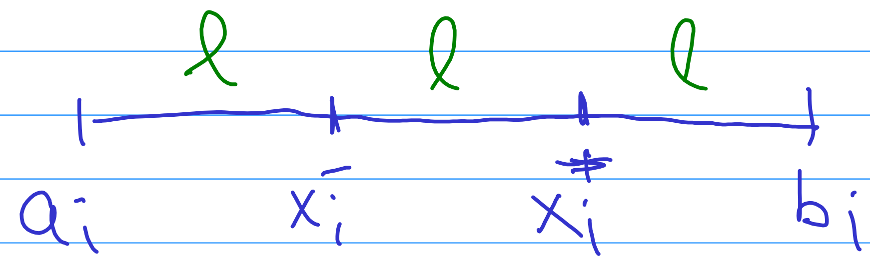
$$\log r_{i+1} \approx \log C + p \log r_i$$

graficul va fi o dreaptă

Propunere unor algoritmi concreți

- alegem 2 puncte intermediare

Trisecției



Dacă $f(x_i^-) \geq f(x_i^+)$

$$[a_{i+1}, b_{i+1}] = [x_i^-, b_i]$$

Altfel

$$[a_{i+1}, b_{i+1}] = [a_i, x_i^+]$$

La fiecare iterație lungimea int. este redusă la $2/3$ din lungimea actuală.

r_i = Eroarea$_i$ = lungimea int nr i

$r_{i+1} = \frac{2}{3} r_i \Rightarrow$ Convergență liniară

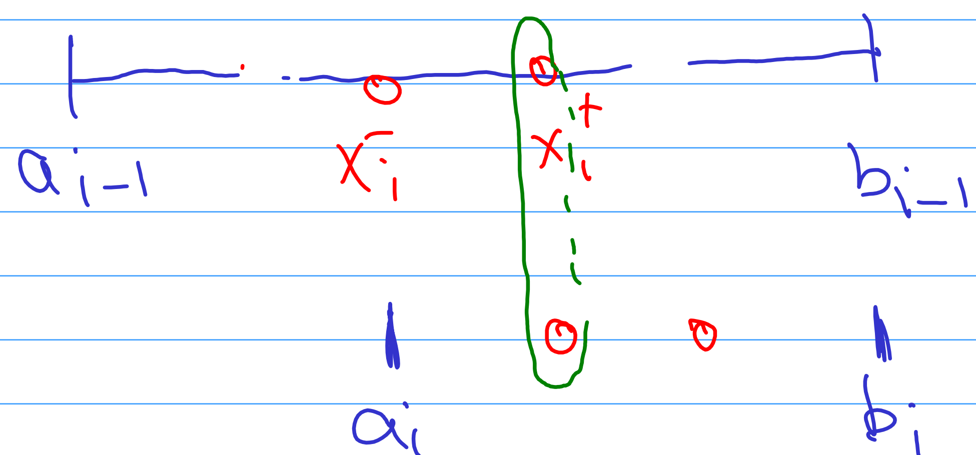
→ $\underline{\text{Raportul de conv:}}$ $\rho = \frac{2}{3}$
(pt nr de iterații)

→ Câștigul pt fiecare evaluare a funcției f este $\sqrt{\frac{2}{3}} \approx 0{,}81$

Putem face doar o evaluare / iterație?

<u>DA</u>

$$|a_{i-1} - x_i^+| = |b_{i-1} - x_i^-| = \frac{F_{N-i}}{F_{N-i+1}} |a_{i-1} - b_{i-1}|$$



• Suprapunere între un pct interm pt iterația i și iterația $i+1$.

<u>Fibonacci search:</u> $\frac{\eta_{i+1}}{\eta_i} \approx \lambda^{-1} = \frac{2}{1+\sqrt{5}} \approx 0{,}61$

→ o singură evaluare de funcție pe iterație.

Fibonacci: $F_{n+1} = F_n + F_{n-1}$
$\forall n \geq 1$

$$F_1 = F_0 = 1$$

1, 1, 2, 3, 5, 8, 13, 21, ...

Calcul algoritmic:

- iterativ → buclă for ce calculează F_n știind pe F_{n-1} și F_{n-2}
- recursivă: ATENȚIE – cost exponențial
  - a nu se utiliza

Eficient:

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\Rightarrow M^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & 0 \end{pmatrix}$$

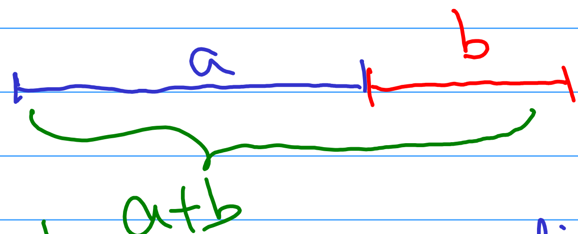
$O(\log n)$

Exponențiere prin ridicare la pătrat.

$$a^n = \begin{cases} a^{n/2} \cdot a^{n/2} & n \text{ par} \\ a^{\frac{n-1}{2}} \cdot a^{\frac{n-1}{2}} \cdot a & n \text{ impar.} \end{cases}$$

Golden search

$$\lambda = \frac{\sqrt{5}+1}{2}$$



$$\lambda = \frac{a}{b} = \frac{a+b}{a}$$ ⇝ ecuație de gradul 2 care îl determină pe λ

$$a^2 = a \cdot b + b^2 \quad | : b$$

$$\underbrace{\left(\frac{a}{b}\right)^2}_{\lambda} = \frac{a}{b} + 1 \Rightarrow \lambda^2 - \lambda - 1 = 0$$

$$\begin{cases} x_i^- = \dfrac{\lambda}{\lambda+1} a_{i-1} + \dfrac{1}{\lambda+1} b_{i-1} \\ x_i^+ = \dfrac{1}{\lambda+1} a_{i-1} + \dfrac{\lambda}{\lambda+1} b_{i-1} \end{cases}$$

$$\text{Eroarea}_{i+1} = \frac{1}{\lambda} \text{Eroarea}_i$$

$$\boxed{\pi_{i+1} = \frac{1}{\lambda} \pi_i}$$

$$1/\lambda \approx 0{,}61$$

Observații despre erori

1. ε mașină?, cel mai mic nr a.î. $1+\varepsilon > 1$

Numerele sunt reprezentate în "virgulă flotantă".

semn exponent

$X \rightarrow$ mantisa

nr biți 0/1

$$X = 1,\underbrace{\quad\quad}_{\text{mantisă (baza 2)}}{}^{(\text{exponent})} \cdot (-1)^{\text{semn}}$$

precizia depinde de mărimea mantisei

$1 + 10^{-16} = 1$ limita de precizie

$$\begin{array}{l} 1,00\ldots\ldots0 \\ 0,0\ldots\ldots0 \\ \hline 1,0\ldots\ldots0 \end{array}$$

1 se pierde.

Dacă $|a-b| \leq \varepsilon$ mașină atunci programul nu diferențiază cele 2 numere.

În jurul punctului de minim, $f \in C^2$

$$f(x) \approx f(x^*) + \frac{1}{2} f''(x^*)(x-x^*)^2$$

$|x-x^*| \leq 10^{-8}$

$\Rightarrow (x-x^*)^2 < 10^{-16}$

programul nu poate diferenția corect $f(x)$ și $f(x^*)$

$\Rightarrow$ Algoritmul nu poate calcula corect $f(x_i^-)$ și $f(x_i^+)$ când $|x_i^{\pm} - x^*| < \varepsilon$