Minimizzazione di funzioni

Matteo Duranti

matteo.duranti@infn.it

(cfr. [http://www.aip.de/groups/soe/local/numres/bookcpdf/c10-0.pdf (legale?!)]

[http://www.aip.de/groups/soe/local/numres/bookcpdf/c10-1.pdf (legale?!)]

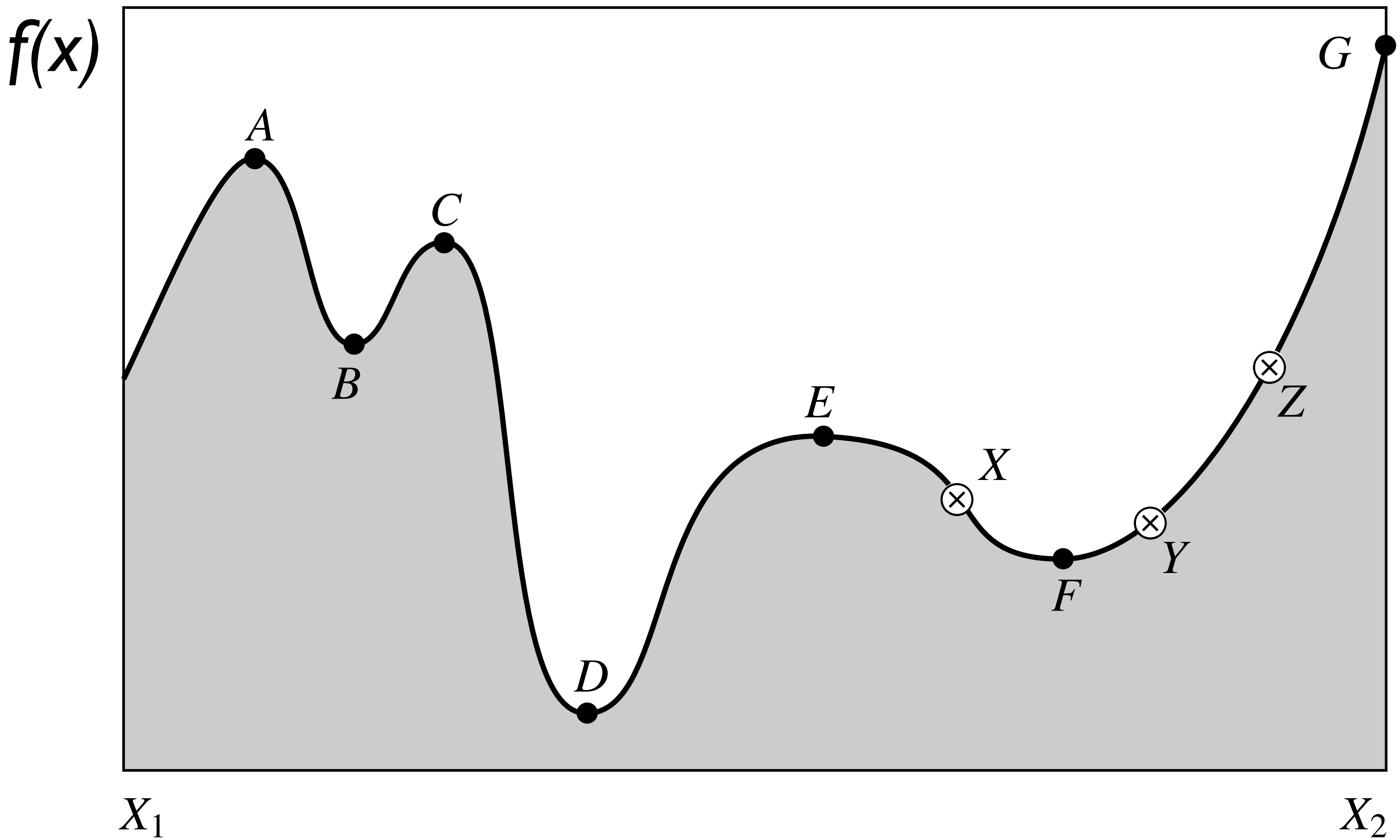
[http://www.aip.de/groups/soe/local/numres/bookcpdf/c10-2.pdf (legale?!)]

http://www.fisica.unipg.it/borromeo/Appunti/FisComp/FisComp2014/pdf/minimi.pdf

http://www.fisica.unipg.it/borromeo/Appunti/FisComp/FisComp2014/pdf/PolinomiERadici.pdf

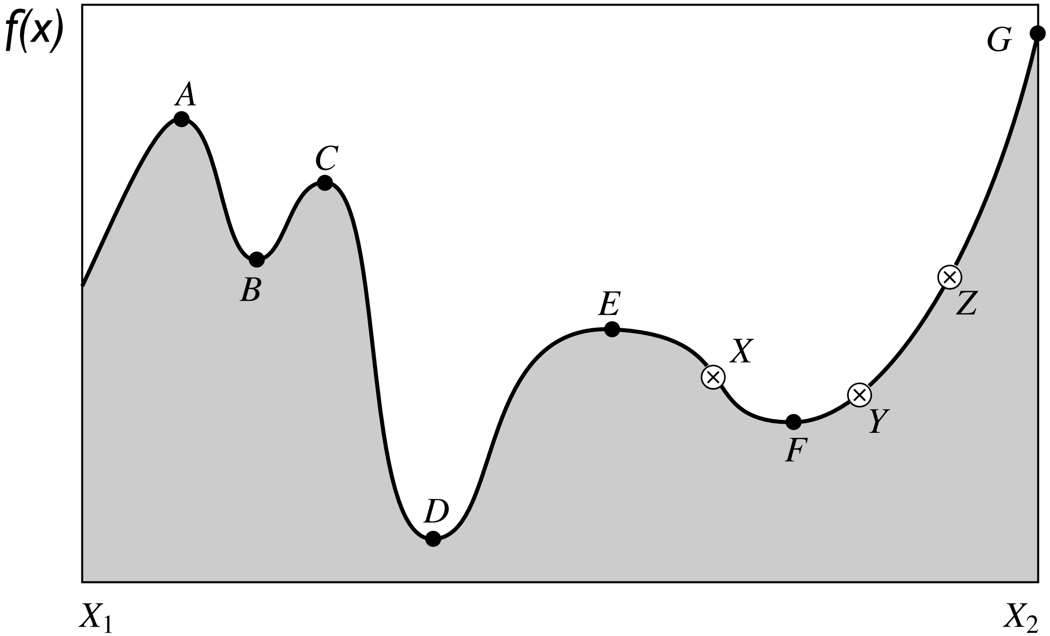
http://www.pspc.unige.it/~fondamenti2/fond2_chp3.pdf)

Minimizzazione di una funzione



Il problema è contettualmente molto semplice: trovare il minimo (o il massimo, $f \rightarrow -f$) di una funzione in un certo intervallo

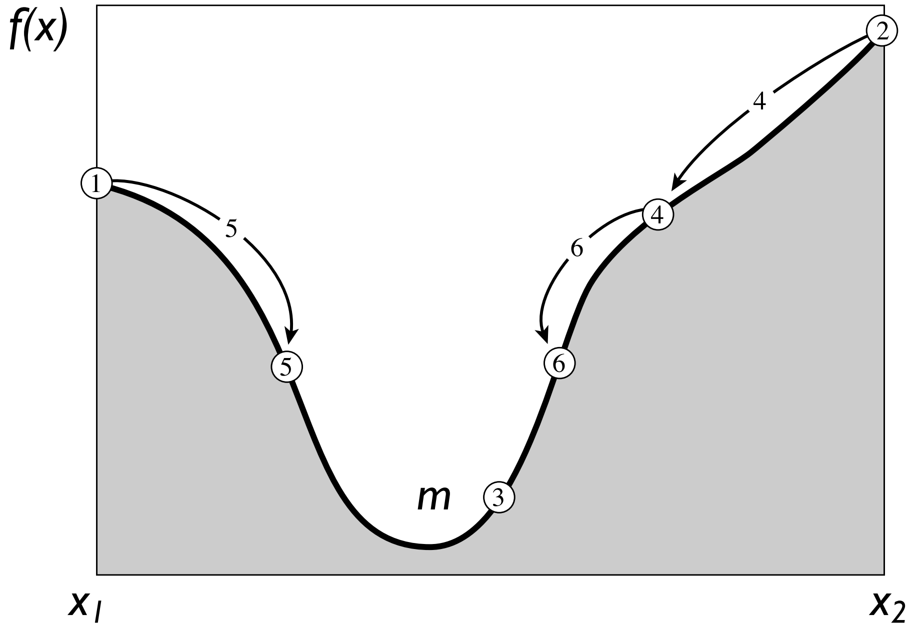
Minimizzazione di una funzione



- *A*, *C* ed *E* sono massimi *locali*
- *B* e *F* sono minimi *locali*
- *D* è un minimo *globale*
- *G* è un massimo *globale* (essendo sul bordo non è detto che $f'(x^2)=0$)
- I punti X, Y e Z "identificano" un minimo (dato che $f_Y<f_X$ e $f_Y<f_Z$)

- Tipicamente siamo interessati ai minimi globali
- Esempi di funzioni da minimizzare
  - energia potenziale in funzione di posizione, rotazione, etc…
  - principi variazionali
  - χ^2 tra dei punti sperimentali e una funzione di fit
  - …
- Un algoritmo efficiente deve trovare il minimo utilizzando il numero minore di valutazioni di *f(x)* possibile

Minimizzazione di una funzione



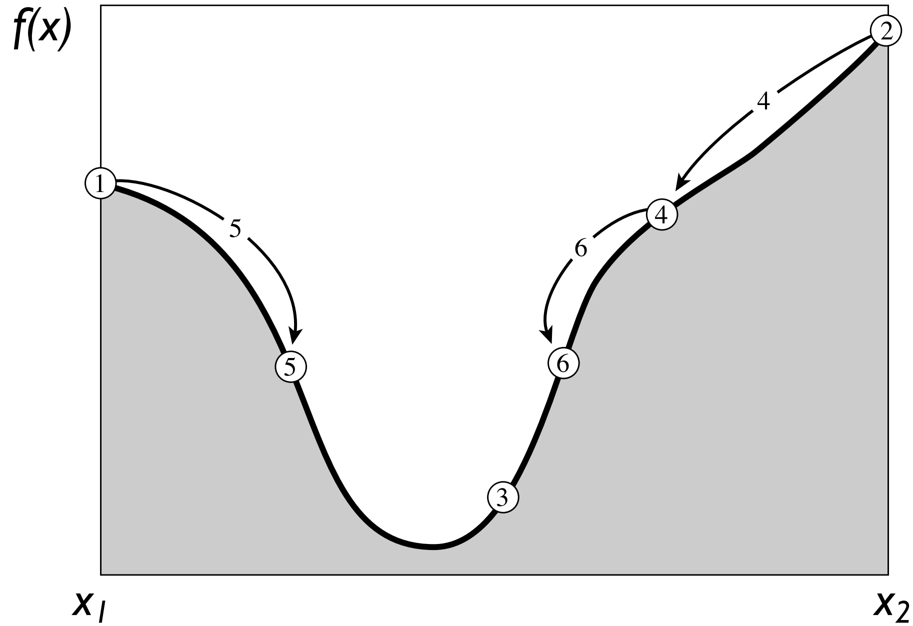
Concettualmente è "molto semplice":
si scansiona $[x_1, x_2]$ alla ricerca di un x_m tale che $f(x_m) \leq f(x)$ $\forall x$ in $[x_1, x_2]$

Il problema, esattamente come per l'integrazione o la risoluzione di ODE, è fare la scansione:

- con il minimo numero di step necessario (che precisione si può/vuole ottenere?)
- con il minor numero di valutazioni di *f(x)* possibile

→ va scelta una strategia per la selezione degli intervalli da scansionare

Minimizzazione di una funzione



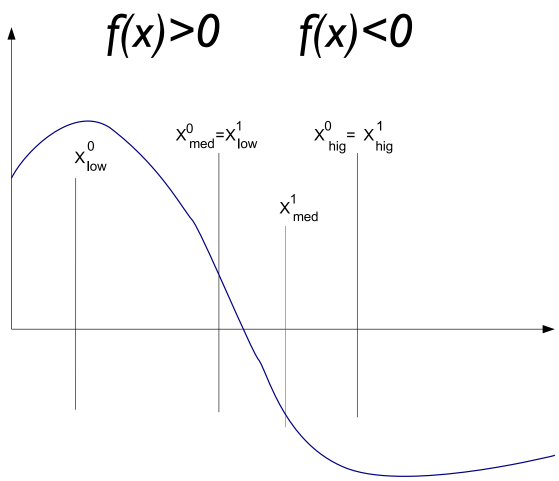
Se conosco tre punti, *[A,B,C]=[1,2,3]* (ma anche *[1,2,4]*, *[1,2,5]*, *[1,2,6]* o *[5,6,3]*, *[5,4,3]*, *…)* tali che:

$x_A < x_C < x_B$ con $f(A) > f(C)$ e $f(B) > f(C)$

allora <u>ho trovato un minimo</u> (locale, i.e. in $[x_A, x_B]$)

Parentesi: ricerca della radice di una funzione

Bisezione



Se in $[x_{low}, x_{hig}]$ esiste una radice e la voglio calcolare con precisione ε suppongo che:

$$f(x_{low}) < 0 \quad \text{e} \quad f(x_{hig}) > 0$$

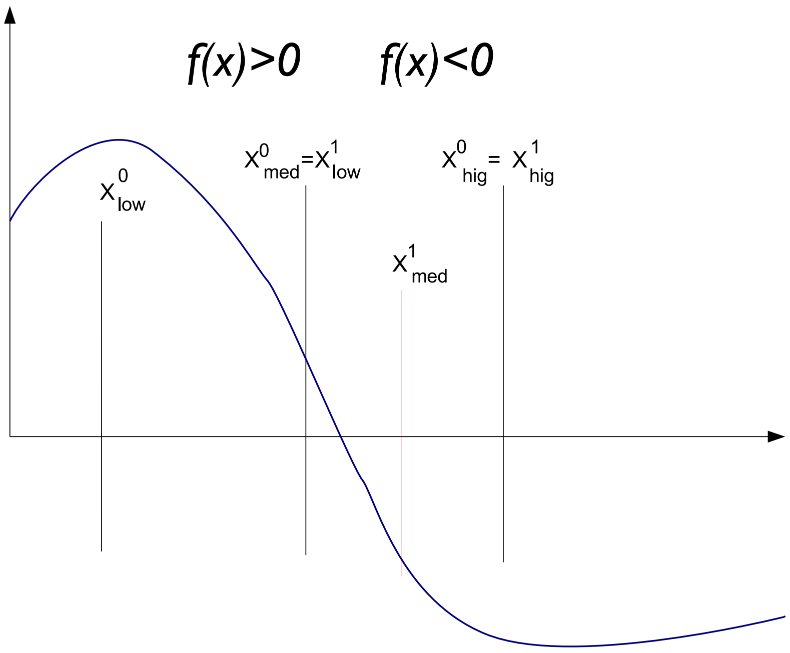
(o viceversa).

Prendendo allora il punto medio

$$x_{med} = \frac{1}{2} \cdot (x_{low} + x_{hig})$$

e calcolo $f(x_{med})$.

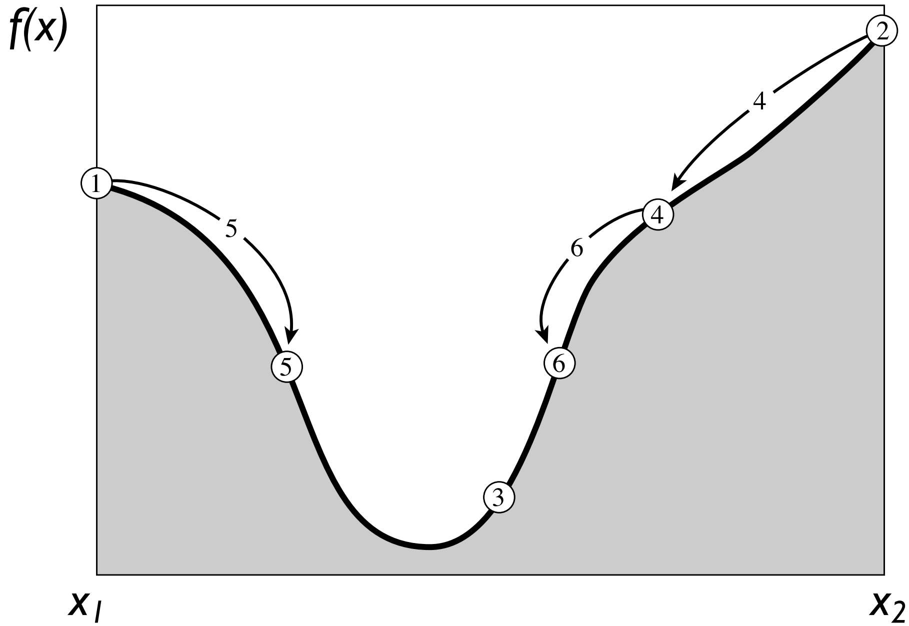
Parentesi: ricerca della radice di una funzione



- Se $|x_{hig} - x_{low}| < \varepsilon$ termino il ciclo;
- se $f(x_{med})$ e $f(x_{low})$ hanno lo stesso segno ;
  1. $[x_{med}, x_{hig}]$ come nuovo intervallo;
  2. $\frac{1}{2} \cdot (x_{med} + x_{hig})$ è il nuovo punto medio;
- Se $f(x_{med})$ e $f(x_{hig})$ hanno lo stesso segno;
  1. $[x_{low}, x_{med}]$ come nuovo intervallo;
  2. $\frac{1}{2} \cdot (x_{low} + x_{med})$ è il nuovo punto medio;
- ripeto il ciclo;

La precisione dopo N passi è $\frac{1}{2^N}$ volte l'intervallo iniziale.

Minimizzazione di una funzione



Se conosco tre punti, *[A,B,C]* tali che:

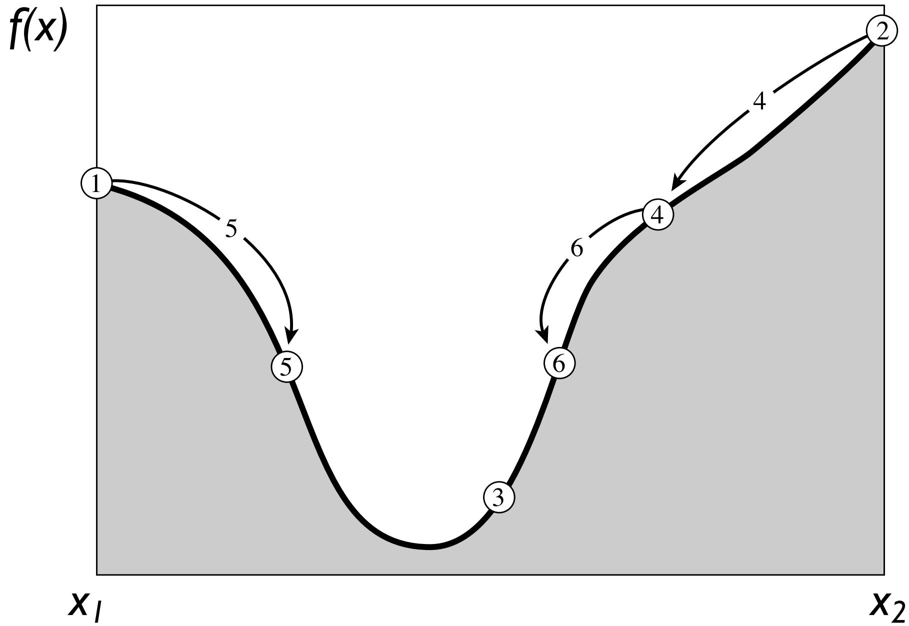
$x_A < x_C < x_B$ con $f(A) > f(C)$ e $f(B) > f(C)$

allora <u>ho trovato un minimo</u> (locale, i.e. in $[x_A, x_B]$),

Confronto con la ricerca delle radici di una funzione:

- nella ricerca della radice di una funzione sono bastati <u>2</u> punti per capire quale fosse "prima" della radice e quale fosse "dopo".
- nella minimizzazione, invece, servono <u>3</u> punti

Minimizzazione di una funzione



Se conosco tre punti, *[A,B,C]* tali che:

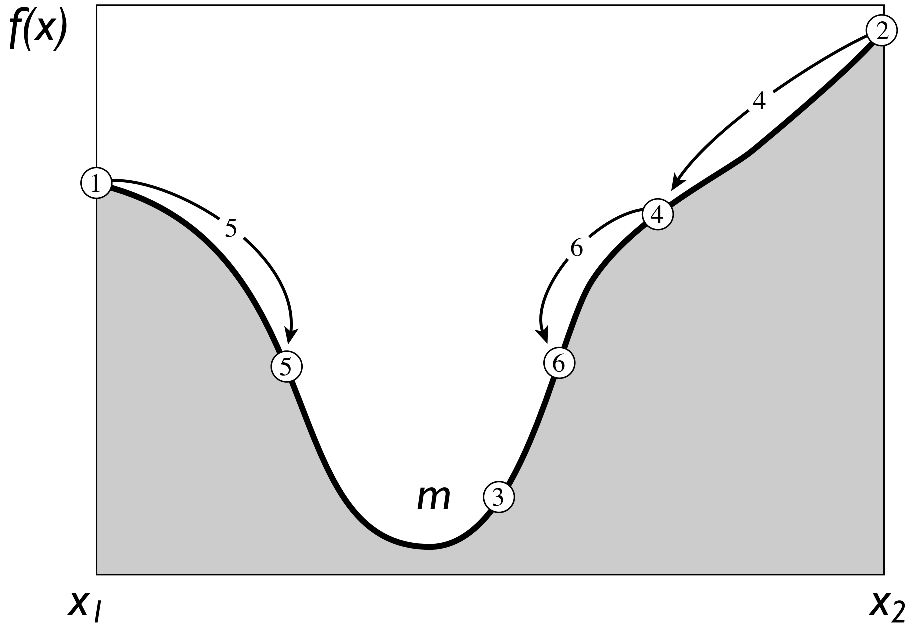
$x_A < x_C < x_B$ con $f(A) > f(C)$ e $f(B) > f(C)$

allora <u>ho trovato un minimo</u> (locale, i.e. in $[x_A, x_B]$),

Se parto da *[1,2,3]* poi utilizzo 5 e 4, restringendomi a *[5,4,3]*, dal momento che $f(5)>f(3)$ e $f(4)>f(3)$ (devono essere vere entrambe: 3 punti!).

Poi posso ulteriormente restringermi a *[5,6,3]*, dal momento che $f(5)>f(3)$ e $f(6)>f(3)$

Minimizzazione di una funzione - precisione

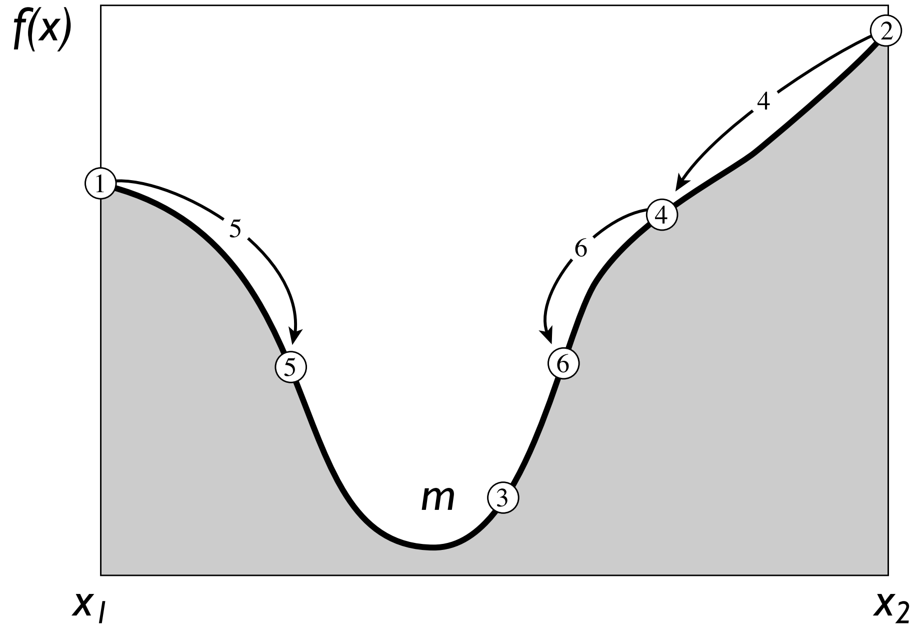


Uno potrebbe pensare che il punto di minimo, *m*, può essere trovato restringendosi ad un intervallo del tipo

$$(1-\varepsilon)b < b < (1+\varepsilon)b$$

dove ε dipende dalla precisione della propria macchina / del proprio codice ($\sim 10^{-8}$ per un `float` e $\sim 10^{-15}$ per un `double`).

Minimizzazione di una funzione - precisione



In prossimità del minimo vale l'approssimazione:

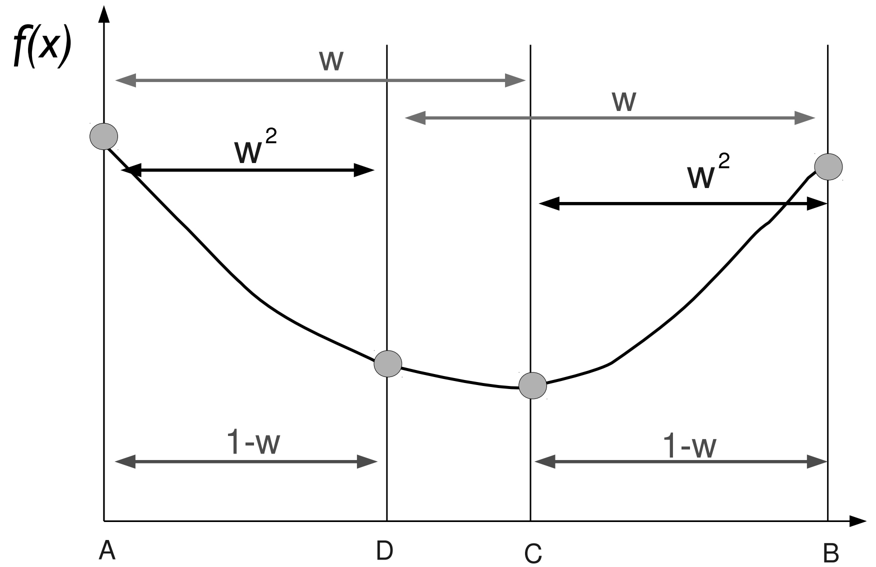
$$f(x) \approx f(x_0) + \frac{1}{2} \cdot f''(x_0) \cdot (x - x_0)^2$$

se $f(x)$ è nota con precisione ε (i.e. $f(x) \sim f(x_0)$) allora:

$$||x - x_0|| = \sqrt{2\varepsilon / f''(x_0)}$$

non senso cercare la *x* del minimo con una precisione migliore di 10^{-4} per un `float` e 10^{-8} per un *double*…

Minimizzazione di una funzione – ricerca aurea



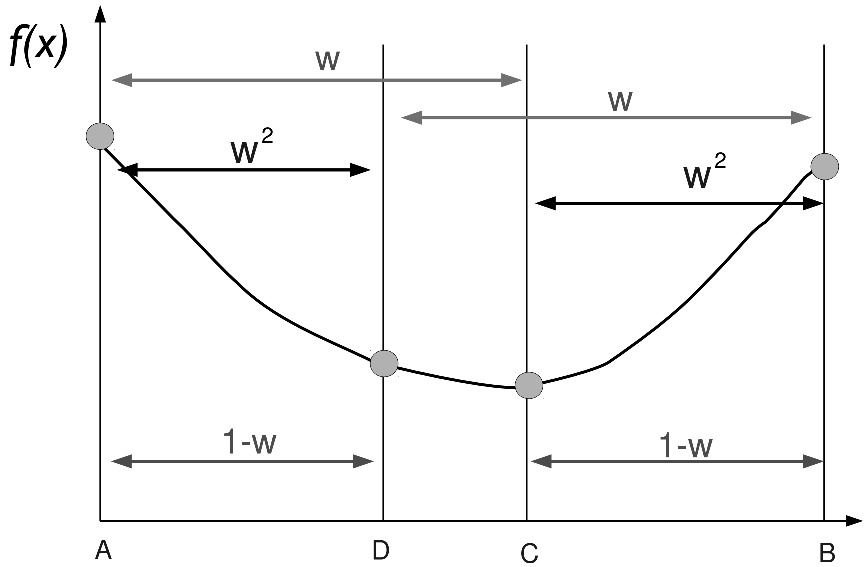
Abbiamo ristretto il nostro intervallo a [A,B] e sappiamo che in C vale:

$$x_A < x_C < x_B \qquad \text{con} \qquad f(A) > f(C) \text{ e } f(B) > f(C)$$

(la procedura che ora "svilupperemo" in realtà la potremo usare, da subito, anche per scegliere *C*)

- assumiamo per semplicità di notazione che $x_B - x_A = 1$
- assumiamo che *C* sia "oltre la metà", cioè che disti ω da *A* e 1-ω da *B* (l'altro caso è ovviamente identico ma speculare)

Minimizzazione di una funzione – ricerca aurea



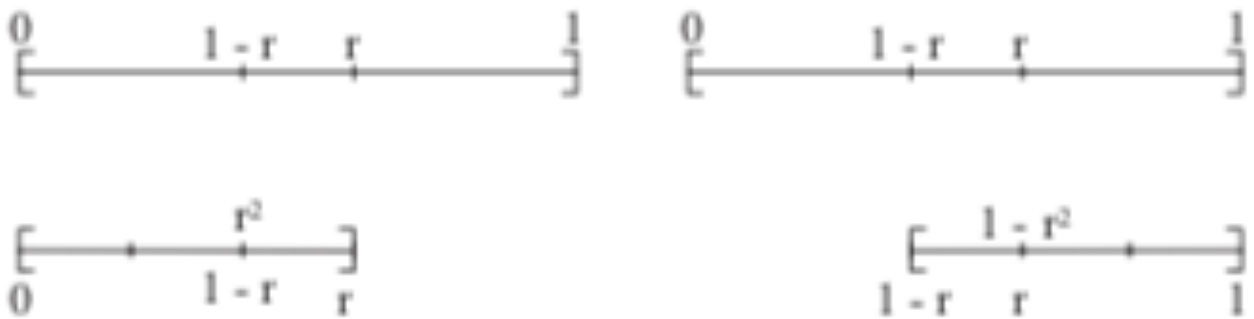
Scelgo un nuovo punto, *D*, tale che $AC = DB = \omega\ AB = \omega$

Il minimo adesso può stare in *[A,C]* o in *[D,B]*. Questo ci viene detto da una, singola, nuova valutazione della funzione, *f(D)*.

Assumiamo che il minimo sia in *[A,C]* (l'altro caso, quello in figura tra l'altro, al solito, è speculare ma identico).

Data la scelta di *D*, il nuovo tripletto *[A,C,D]* è identico (ma scalato) al vecchio *[A,B,C]*

Minimizzazione di una funzione – ricerca aurea



Poiché i tripletti *[A,B,C]* e *[A,C,D]* sono praticamente identici, deve valere la proporzione:

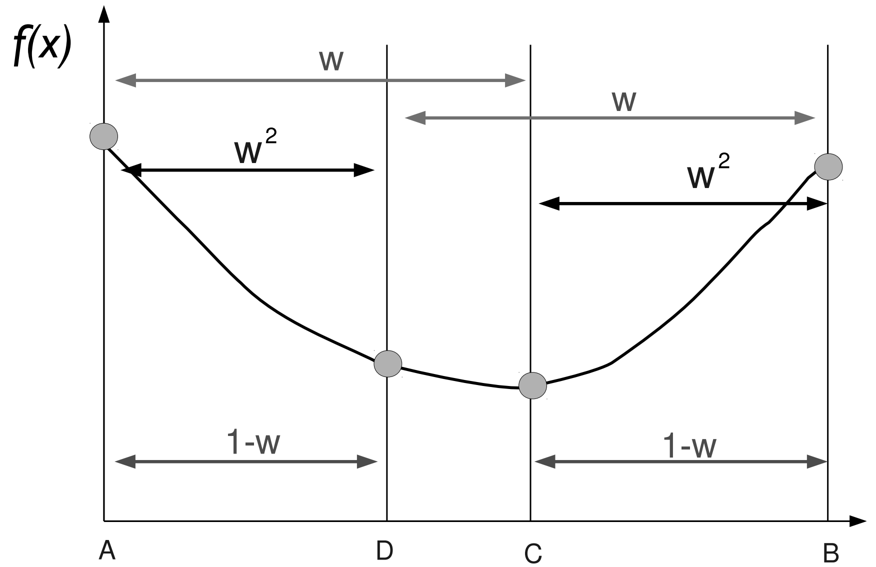
$$1-\omega : \omega = \omega : 1$$

cioè:

$$\omega^2 = 1-\omega$$

$\omega = (sqrt(5) - 1) / 2 \sim 0.618$, cioè la sezione aurea

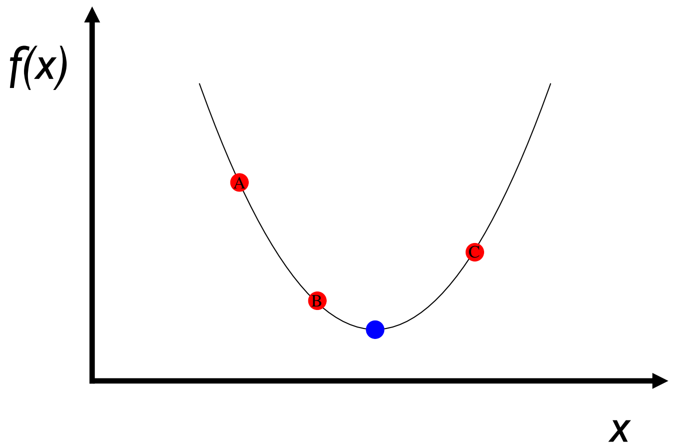
Minimizzazione di una funzione – ricerca aurea



In sostanza:

- abbiamo ridotto l'intervallo di ricerca di quasi un fattore 2 (1.618 = 1/0.618)
- identifichiamo il nuovo intervallo, fra due, con una sola nuova valutazione di *f(x)*
- il restringimento dell'intervallo lo ripetiamo iterativamente (tenendo sempre a mente $||x - x_0|| = \sqrt{2\varepsilon / f''(x_0)}$)
- ad ogni nuovo intervallo *D*' costituisce la nostra stima del punto di minimo

Minimizzazione di una funzione – approssimazione iperbolica



Assumiamo che vicino al minimo la funzione possa essere ben approssimata come una parabola:

$$f(x) \approx P + Q \cdot (x - x_0)^2$$

trovare x_0 significa trovare il minimo (se l'approssimazione è valida). Con tre valutazioni di *f(x)* possiamo scrivere 3 equazioni in 3 incognite, x_0, *P* e *Q*:

$$\begin{aligned} P + Q \cdot (x_A - x_0)^2 &= f_A \\ P + Q \cdot (x_B - x_0)^2 &= f_B \\ P + Q \cdot (x_C - x_0)^2 &= f_C \end{aligned}$$

Minimizzazione di una funzione – approssimazione iperbolica

Sottraendo le equazioni a due a due mi libero di *P*:

$$Q \cdot \left[(x_A - x_0)^2 - (x_B - x_0)^2\right] = f_A - f_B$$
$$Q \cdot \left[(x_B - x_0)^2 - (x_C - x_0)^2\right] = f_B - f_C$$

con il rapporto elimino *Q*:

$$\frac{(x_A - x_0)^2 - (x_B - x_0)^2}{(x_B - x_0)^2 - (x_C - x_0)^2} = \frac{f_A - f_B}{f_B - f_C}$$

cioè

$$\frac{x_A^2 + x_0^2 - 2x_A x_0 - x_B^2 - x_0^2 + 2x_B x_0}{x_B^2 + x_0^2 - 2x_B x_0 - x_C^2 - x_0^2 + 2x_C x_0} = \frac{f_A - f_B}{f_B - f_C}$$

$$\frac{x_A^2 - x_B^2 + 2(x_B - x_A)x_0}{x_B^2 - x_C^2 + 2(x_C - x_B)x_0} = \frac{f_A - f_B}{f_B - f_C}$$

$$\left(x_A^2 - x_B^2 + 2(x_B - x_A)x_0\right)(f_B - f_C) \quad =$$
$$\left(x_B^2 - x_C^2 + 2(x_C - x_B)x_0\right)(f_A - f_B)$$

Minimizzazione di una funzione – approssimazione iperbolica

che quindi ci da la formula per il punto x_0, che pensiamo essere il punto di minimo:

$$x_0 = \frac{1}{2} \cdot \frac{\left(x_A^2 - x_B^2\right)(f_B - f_C) + \left(x_B^2 - x_C^2\right)(f_B - f_A)}{[(x_C - x_B)(f_A - f_B) + (x_B - x_A)(f_C - f_B)]}$$

Da notare:

- il metodo NON è iterativo
- il metodo funziona solo se ci siamo già ristretti ad un intervallo ridotto, in cui è presente il minimo e in cui possa valere l'approssimazione parabolica

→ tipicamente si utilizza la ricerca aurea su larghi intervalli, quando la funzione presenta diversi minimi locali e, una volta ristretto l'intervallo si utilizza l'approssimazione parabolica per convergere velocemente al minimo