

04 - Zeri di funzioni

Approssimazioni di zeri di funzioni reali

Sia data una funzione $f : \mathbb{R} \Rightarrow \mathbb{R}$ continua.

Determinare x tale che $f(x) = 0$

Tali valori sono solitamente chiamati **zeri** o **radici** della funzione f

Radice Semplice

Se $\alpha \in \mathbb{R}$ si dice **radice semplice** di f se:

- $f(\alpha) = 0$
 - $f'(\alpha) \neq 0$
-

Radice Multipla

Se $\alpha \in \mathbb{R}$ si dice **radice multipla** di f di molteplicità m se:

- $f(\alpha) = 0$
- $f'(\alpha) = 0$
- ...
- $f^{m-1}(\alpha) = 0$
- $f^m(\alpha) \neq 0$

La **molteplicità della radice** consiste quindi nel numero di derivate che si annullano.

Come fare?

Riduzione intervallo (separazioni delle radici)

Ridurre l'intervallo della funzione in modo che ci sia un solo zero di funzione in quell'intervallo $[a, b]$.

Si può fare graficando la funzione e cercando l'intervallo ad occhio.

Studio condizionamento del problema

Problema: determinare $\alpha \in \mathbb{R}$ t.c. $f(\alpha) = 0$

Problema perturbato: determinare $\alpha^* = \alpha + \delta$ t.c. $f^*(\alpha^*) = 0$ dove

- $f^* = f + \epsilon g$
- $g : \mathbb{R} \Rightarrow \mathbb{R}$
- ϵ è la perturbazione

Dove f è il dato del problema mentre α è il risultato, quindi δ è la perturbazione del risultato mentre ϵ è la perturbazione sui dati (ovvero la perturbazione sulla funzione in input)

Indice di condizionamento

Tramite lo sviluppo di Taylor otteniamo che l'indice di condizionamento risulta:

$$K := \left(\frac{1}{|f'(\alpha)|} \right)$$

Se $|f'(\alpha)|$ è molto piccolo, allora il problema è mal condizionato.

Viceversa il problema risulta ben condizionato e $f^*(x) = 0$ ha una radice α^* che non differisce troppo da α .

Algoritmi iterativi

Viene dato $x_0 \in \mathbb{R}$ valore di innesco.

Il metodo genera a partire da x_0 una successione di valori $\{x_i\}_{i \geq 0}$ che converge ad α .

$$\lim_{i \rightarrow +\infty} x_i = \alpha$$

Due tipi di algoritmi:

- **a convergenza globale** = non dipende dal valore di innesco x_0 (converge sempre ad α)
- **a convergenza locale** = dipende fortemente dal valore di innesco x_0 (ne servirà uno molto vicino ad α altrimenti non converge)

Un tipico algoritmo lavora quindi nel seguente modo:

```
x0 = input()
x = x0
while(condizioneDiArresto(x)):
    x = calcolaSuccessivo(x)
```

Criteri di arresto

Definisco l'errore assoluto del passo i-esimo come:

$$e_i = |x_i - \alpha| < tolleranza$$

Primo criterio

Tuttavia non si conosce α quindi si utilizza la differenza rispetto al valore successivo dato che esso sarà più vicino ad α .

$$e_i = |x_i - x_{i+1}| < tol$$

Secondo criterio

Sviluppandolo ulteriormente portando questo errore ad un tipo relativo:

$$e_i = \left(\frac{|x_i - x_{i+1}|}{|x_{i+1}|} \right) < tol$$

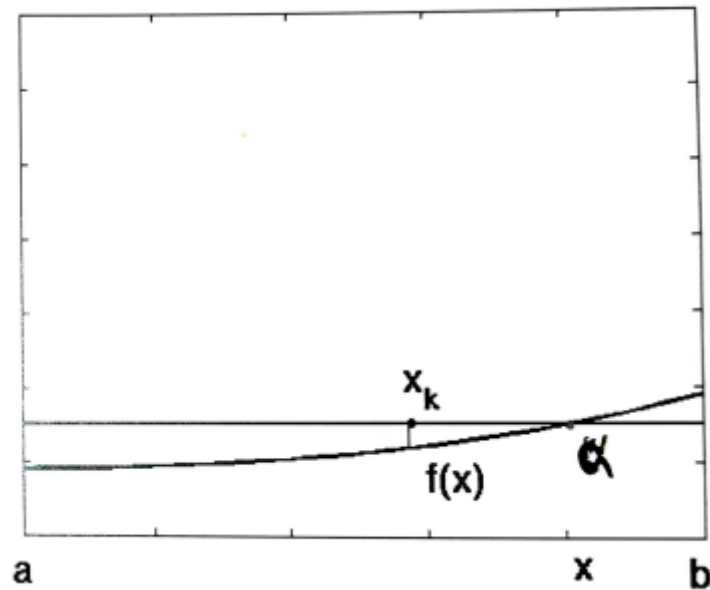
Terzo criterio

Un altro ipotetico criterio di arresto potrebbe essere il **controllo sul residuo**, ovvero:

$$|f(x_i)| < tol$$

Tuttavia quest'ultimo criterio non va mai utilizzato da solo poichè potrebbe dare una falsa condizione di uscita.

Esempio:



Il gap tra 0 e $f(x_i)$ è molto piccolo tuttavia α è molto lontano.

Stima numerica dell'ordine di convergenza del metodo iterativo

Richiami

Una successione $\{x_i\}_{i \geq 0}$ convergente ad un limite α si dice **convergente di ordine p** se esistono $p \geq 1$ e $c > 0$ t.c.

$$\lim_{k \rightarrow +\infty} \left(\frac{|e_{k+1}|}{|e_k|^p} \right) = C$$

dove $e_k := x_k - \alpha$

In altre parole, per valori di k grandi risulta

$$|e_{k+1}| \simeq C|e_k|^p$$

$$|e_{k+2}| \simeq C|e_{k+1}|^p$$

e di conseguenza:

$$\left(\frac{|e_{k+2}|}{|e_{k+1}|}\right) = \left(\frac{|e_{k+1}|}{|e_k|}\right)^p$$

applicando il logaritmo:

$$p = \left(\frac{\log\left(\frac{|e_{k+2}|}{|e_{k+1}|}\right)}{\log\left(\frac{|e_{k+1}|}{|e_k|}\right)}\right)$$

e infine passando ad x :

$$p = \left(\frac{\log\left(\frac{|x_{k+2}-x_{k+3}|}{|x_{k+1}-x_{k+2}|}\right)}{\log\left(\frac{|x_{k+1}-x_{k+2}|}{|x_k-x_{k+1}|}\right)}\right)$$

dove x_{k+3} è l'ultima iterata

p prende il nome di **ordine di convergenza** mentre la costante C prende il nome **fattore di convergenza**

Per i metodi che studiamo abbiamo che:

- $p = 1$ ($C < 1$) **convergenza lineare**
- $1 < p < 2$ **convergenza superlineare**
- $p = 2$ **convergenza quadratica**

Metodo di bisezione ($p = 1$, $C = \frac{1}{2}$)

PRO:

- convergenza globale
- è l'unico metodo in cui posso determinare a priori il numero di iterazioni necessario

CONTRO:

- lento (C non è vicino a 0)

COSTO COMPUTAZIONALE = numero di iterazioni eseguite + 2

CONDIZIONI:

- f continua in $[a, b]$
- $f(a) * f(b) < 0$ (valori discordi agli estremi)

Procedimento

Suppongo $a < 0, b > 0$

1. mi calcolo il punto medio dell'intervallo $[a, b]$ denominandolo x_1 , se $f(x_1) = 0$ ho finito.
2. scelgo il sottointervallo di $[a, b]$ dove continuano a valere le condizioni
 - $x_1 < 0 \Rightarrow [x_1, b]$
 - $x_1 > 0 \Rightarrow [a, x_1]$
3. torna al punto 1

Condizione di arresto

$$|b_i - a_i| \geq tol + eps \times \max(\{|a_i|, |b_i|\})$$

Output

Ultimo punto medio calcolato

Numero di iterazioni

$$n_\epsilon \geq \log_2\left(\frac{b-a}{\epsilon}\right) = 3.3 \times \log_{10}\left(\frac{b-a}{\epsilon}\right)$$

Pseudocodice

```
def bisezione(f, a, b):
    middle = -1
    fa = f(a) # Iterazione 1
    fb = f(b) # Iterazione 2
    while(abs(b - a) >= tol + eps*max([abs(a), abs(b)])):
        # non usiamo (a + b) / 2 poichè non è stabile
        middle = a + (b - a) / 2
        fmiddle = f(middle) # Iterazione
        if sign(fmiddle) != sign(fa) < 0:
            b = middle
            fb = fmiddle
        elif sign(fmiddle) != sign(fb) < 0:
            a = middle
            fa = fmiddle
    return middle
```

Metodo di falsa posizione (regula falsi)

Dati i punti a, b si traccia la retta tra di essi.

Di questa retta si trova l'intersezione con l'asse x denominato x_i e calcoliamo $f(x_i)$ per poi procedere come il metodo di bisezione.

Cambia solo il modo di trovare il punto medio.

$$x_i = a_i - f(a) \times \left(\frac{b - a}{f(b) - f(a)} \right)$$

È leggermente più veloce di Bisezione

Pseudocodice

```
def regula_falsi(f, a, b):
    middle = -1
    fa = f(a) # Iterazione 1
    fb = f(b) # Iterazione 2
    while(abs(b - a) >= tol + eps*max([abs(a), abs(b)])):

        #####
        middle = a - fa*((b-a)/(fb-fa)) ## Unica Differenza
        #####

        fmiddle = f(middle) # Iterazione
        if sign(fmiddle) != sign(fa) < 0:
            b = middle
            fb = fmiddle
        elif sign(fmiddle) != sign(fb) < 0:
            a = middle
            fa = fmiddle
    return middle
```

Nuova metodologia

- Scegliere un valore di innesco x_0
- Costruire una successione di valori x_i dove x_{i+1} risulta essere l'ascissa del punto di intersezione tra l'asse x e la retta di equazione $y = f(x_i) + m_i(x - x_i)$

$$x_{i+1} := x_i - \left(\frac{f(x_i)}{m_i} \right) \quad m_i \neq 0$$

Da questa metodologia derivano tutti gli altri metodi che andremo a studiare, i quali si differenzieranno solamente per la scelta di m_i :

- metodo delle **corde** ($m_i = m = \text{costante}$)
 - metodo delle **secanti** ($m_i = \left(\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \right)$)
 - metodo di **Newton** ($m_i = f'(x_i)$)
-

Metodo delle corde

$$m_i = m = \text{costante}$$

di solito si sceglie $m_i = f'(x_0)$

Abbiamo quindi che al passo i -esimo:

$$x_{i+1} := x_i - \left(\frac{f(x_i)}{m_i} \right)$$

Teorema

Sia $f : [a, b] \Rightarrow \mathbb{R}$ tale che $f(\alpha) = 0$ per $\alpha \in [a, b]$.

Condizioni sufficienti per la convergenza del metodo delle corde sono:

- $f'(x) \neq 0 \quad \forall x \in [a, b]$
- $mf'(x) > 0 \quad \forall x \in [a, b]$
- $m > \frac{1}{2} \times \max_{x \in [a, b]} f'(x)$

Metodo delle secanti ($p = 1.6$)

$$m_i = \left(\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \right)$$

Abbiamo quindi che al passo i -esimo:

$$x_{i+1} := x_i - f(x_i) \times \left(\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \right)$$

N.B. Come valore di innesco in questo caso abbiamo bisogno di due valori (x_i, x_{i-1})

Il metodo delle secanti può essere più veloce della regola falsi, ma non converge sempre.

La convergenza è garantita se le approssimazioni iniziali sono 'abbastanza vicine' alla radice α : convergenza locale.

In tal caso **la convergenza è superlineare**

($p = 1.618$).

Metodo di Newton ($p = 2$)

$x_0 = \text{input}$

$$m_i = f'(x_i)$$

quindi al passo i -esimo avremo:

$$x_{i+1} = x_i - \left(\frac{f(x_i)}{f'(x_i)} \right)$$

L'aumento del costo computazionale è compensato dal fatto che la convergenza è di ordine superiore al primo.

Sotto l'ipotesi che α è radice semplice e $f \in C^3[a, b]$, la convergenza è quadratica ($p = 2$)

Teoremi di convergenza del metodo di Newton

Se f ha concavità fissa in $[a, b]$, è possibile stabilire un criterio di scelta dell'approssimazione iniziale x_0 che garantisce la convergenza del metodo di Newton.

Data una funzione f continua e convessa in $[a, b]$ con $f(a)f(b) < 0$, si dice **estremo di Fourier** di $[a, b]$ l'estremo verso cui f rivolge la convessità.

N.B: Se esiste f'' , allora l'estremo di Fourier di $[a, b]$ è a o b a seconda che $f(a)f''(a) > 0$ oppure $f(b)f''(b) > 0$.

In soldoni

Si sceglie l'estremo di Fourier tracciando una retta tra a e b e scegliendo la x corrispondente ad a o b che si trova nello stesso semipiano della funzione.

Teorema 1

Se $f : [a, b] \rightarrow \mathbb{R}$ soddisfa le seguenti ipotesi:

- $f(a)f(b) < 0$
- f, f', f'' sono continue in $[a, b]$, ossia $f \in C^2[a, b]$
- $f'(x) \neq 0, \forall x \in [a, b]$
- $f''(x) \neq 0, \forall x \in [a, b]$

e se l'approssimazione iniziale di x_0 è scelta coincidente con l'estremo di Fourier dell'intervallo $[a, b]$, allora il metodo di Newton definisce una successione monotona e convergente all'unica radice.

La convergenza è **superlineare**

Se $f \in C^3[a, b]$ allora la convergenza è **quadratica** e vale inoltre che **l'ordine di convergenza** è:

$$C = \lim_{x \rightarrow +\infty} \left(\frac{|x_{i+1} - \alpha|}{(x_i - \alpha)^2} \right) = \left(\frac{f''(\alpha)}{2f'(\alpha)} \right)$$

Metodi iterativi di punto fisso

x_0 in input

$$x_{i+1} = g(x_i), i \geq 0$$

Se si sceglie:

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

allora possiamo inquadrare il metodo di Newton come un particolare metodo iterativo di punto fisso.

Condizione di arresto solo sull'incremento, non va bene il residuo perchè tende ad α

Teorema di convergenza globale

Si consideri la successione $x_{i+1} = g(x)$, $i = 0, 1, 2, \dots$ con x_0 assegnato.

Si supponga che:

1. $g : [a, b] \rightarrow [a, b]$
2. $g \in C^1[a, b]$
3. $\exists C < 1$ t.c. $|g'(x)| \leq C, \forall x \in [a, b]$

Allora g ha un unico punto fisso $\alpha \in [a, b]$ e la successione converge ad α per scelta di $x_0 \in [a, b]$.

N.B: NON LO USEREMO MAI (F)

Teorema di convergenza locale (Ostrowski)

Sia α un punto fisso di $g \in C^1[\alpha - \rho, \alpha + \rho], \rho > 0$

Se:

$$|g'(x)| < 1, \forall x \in [\alpha - \rho, \alpha + \rho]$$

allora $\forall x_0 \in [\alpha - \rho, \alpha + \rho]$ la successione delle iterate generate da g è tale che:

1. $x_i \in [\alpha - \rho, \alpha + \rho], \forall i \geq 1$
 2. $\lim_{i \rightarrow +\infty} x_i = \alpha$ unico punto fisso di g
-

Ordine di convergenza del metodo di punto fisso

Sia $g \in C^p(I)$

Se per un $x_0 \in I$ la successione delle iterate x_i è convergente e se:

- $g(\alpha) = \alpha$
- $g'(\alpha) = 0$
- ...
- $g^{p-1}(\alpha) = 0$
- $g^p(\alpha) \neq 0$

allora:

$$C = \left(\frac{|g^{(p)}(\alpha)|}{p!} \right)$$

ossia il metodo ha ordine di convergenza p

Metodo di Newton modificato

Se utilizzo Newton su radici semplici ha ordine di convergenza $p = 2$, mentre se lo utilizzo su radici multiple perde la sua efficacia perchè l'ordine di convergenza diventa $p = 1$.

Tuttavia se lo modifichiamo nel seguente modo:

$$x_{i+1} = x_i - m \times \left(\frac{f(x_i)}{f'(x_i)} \right)$$

dove **m** è la **molteplicità della radice** allora otteniamo di nuovo un ordine di convergenza $p = 2$
