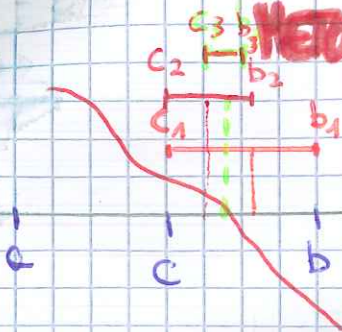


METODO BISEZIONE



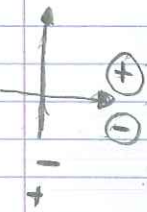
$c = \text{punto medio tra } [a, b]$

importante

• devo sempre stare tra l'intervallo + -

• Se $f(\text{medio}) = 0$ significa che

medio = radice cercata



$$f(a) \cdot f(b) < 0 \Rightarrow [a, b]$$

$$c = \frac{a+b}{2}$$

$$\text{se } f(a) \cdot f(b) < 0 \quad [a, c]$$

altrimenti la radice si trova in $[c, b]$

$$[a_0, b_0] \supseteq [a_1, b_1] \supseteq \dots \supseteq [a_m, b_m]$$

invariato

$$b_0 = a_0$$

$$b_1 - a_1 = \frac{1}{2} (b_0 - a_0)$$

$$b_2 - a_2 = \frac{1}{2} (b_1 - a_1) = \frac{1}{2^2} (b_0 - a_0)$$

$$b_m - a_m = \frac{1}{2^m} (b_0 - a_0)$$

$$x_m = \frac{a_m + b_m}{2}$$

$$\text{L'errore } E = |x - x_m| \leq \frac{1}{2} |b_m - a_m| \leq \frac{1}{2^m} |b_0 - a_0|$$

errore che commetto ad ogni passo
e al massimo metà dell'intervallo finale

VERO! $E_m < \frac{1}{2^{m+1}}$ $\text{ERRORE} = 0$
 $m \rightarrow \infty$

errore al passo $m+1$

Non vale

$$\frac{|E_{m+1}|}{|E_m|} \leq \frac{1}{2}$$

errore di m

perché se taglio prima
vicino allo zero e poi
più distante

dimostrato che
Non vale

Dimostrata Convergenza

Se eseguo m passi con $m \rightarrow +\infty$
origino un errore pari a zero

Es:

$$\begin{array}{ll} a_0 = -10 & a_1 = -10 \\ b_0 = 14 & b_1 = 2 \\ c_0 = 2 & c_1 = -4 \end{array}$$

$$E_{\text{passo } 0} = |2 - 0| = 2$$

$$E_{\text{passo } 1} = |4 - 0| = 4$$

Quanti passi devo fare per avere una Tolleranza

$$\frac{1}{2^{m+1}} (b_0 - a_0) < \text{Tolleranza}$$

$$\frac{|b_0 - a_0|}{\text{Tolleranza}} < 2^{m+1}$$

$$m > \left\lceil \log_2 \left(\frac{b_0 - a_0}{\text{Tolleranza}} \right) - 1 \right\rceil$$

Calcolato m passi

intero superiore

Velocità di convergenza di ordine p

$$\{x_0\} \rightarrow \{x_1\} \rightarrow \{x_2\}$$

$$\frac{b_0 - a_0}{2} \rightarrow \frac{b_1 - a_1}{2} \rightarrow \frac{b_2 - a_2}{2}$$

$$\lim_{m \rightarrow \infty} \frac{|x_{m+1} - x|}{|x_m - x|^q} = \lim_{m \rightarrow \infty} \frac{|\varepsilon_{m+1}|}{|\varepsilon_m|^q} = C \quad (\neq 0, \neq \infty)$$

Metodo di Bisezione con convergenza lineare

ed ogni passo guadagna una cifra decimale ESATTA

$p=1$ (LINEARE) $\rightarrow$ convergenza lineare

$$\varepsilon_0 = 1$$

$$\varepsilon_m = \frac{1}{10^m} \sum_{m=1}^1$$

$$\varepsilon_2 = \frac{1}{10} \cdot 0,1 = 0,01$$

$$\varepsilon_3 = \frac{1}{10} \cdot 0,01 = 0,001$$

$1 < p < 2 \leftarrow$ convergente superlineare

$p=2$ (convergente QUADRATICA)

$\downarrow$
cioè servono pochi passi del teorema per arrivare alla convergenza (cioè alla precisione macchina)

$$\epsilon_0 = 1$$

$\Downarrow$
molto efficiente per CALCOLARE gli zeri di una radice

$$\epsilon_m = \frac{1}{10^m} \sum_{n=1}^2$$

$$\epsilon_1 = \frac{1}{100} \cdot 0, = 0,1$$

$$\epsilon_2 = 0,001$$

$$\epsilon_3 = 0,00001$$

Infine possiamo dire che il metodo di bisezione converge linearmente nelle AMPIEZZE perché non ha senso parlare soltanto di convergenza purché $\frac{\epsilon_{m+1}}{\epsilon_m} \neq \frac{1}{2}$ NON VALE leggi prima.

PROVA IMPLEMENTAZIONE METODO BISEZIONE (PSEUDOCODICE MATLAB)

PARAMETRI: Funzione, punto A, punto B, TOLLERANZA, MAX_ITERAZIONI

$\rightarrow$ Non si fanno funzioni ricorsive

// TEST ESTREMI

IF $F(a) \cdot F(b) > 0$
ERRORE

END

ITER = 0

AMPIEZZA = $|b - a|$

While AMPIEZZA > TOLLERANZA && ITER <= MAX_ITERAZIONE

ITERAZIONE++

$$X_M = \frac{a+b}{2}$$

if $F(A) \cdot F(X_M) < 0$

B = X_M

else if

A = X_M

End

DIFF. LINEARE e non lineare!

TEST

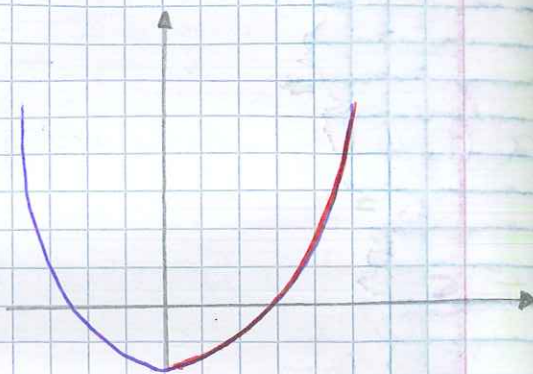
$$f(x) = x^2 - 2$$

$$x_1 = \sqrt{2} = 1,414$$

$$x = \text{BISEZIONE}(f, 0, 2, \text{TOLLERANZA})$$

$$f = \text{inline}('x.^2 - 2', 'x')$$

$$f = (f(x) - x.^2 - 2) \quad \leftarrow \text{per trovare punto medio}$$



ITE	a	b	x _{medio}
1	0	2	1
2	1	2	1,5
3	1	1,5	1,25
4	1,25	1,5	1,375

se $f(a) \cdot f(b) > 0$ in MATLAB `error('`

`)` -> PROGRAMMA
CALCOLO
ESCE

`x = []` restituisce
VUOTO

TEST di ARRESTO dell'ALGORITMO

- ERR. ASSOLUTI
- $|b_m - a_m| < \text{TOLLERANZA}$
 - $|f(x_m)| < \text{TOLLERANZA}$

Dobbiamo richiedere un RANGE di tolleranza opportuna

ES: 1 bon = $1 \cdot 10^5$ Pa ORDINE 10^5
[possiamo chiederne al MAX 10 cifre]
dopo la virgola
Se usiamo in tutto 15 cifre
per rappresentazione il numero

chiedi

Per sapere quante cifre Richiedere dopo la virgola per TOLLERANZA

$$f(x) = ax + b$$

$$E_{\text{ASS}} = |x_A - x^*| \quad x^* = -\frac{b}{a}$$

VALORE
VERO

VALORE MISURATO

$$F(y) = ay + b$$

$$E_{\text{RELATIVO}} = \frac{|x_A - x^*|}{|x^*|} = \left| \frac{x_A}{x^*} - 1 \right| \quad \leftarrow \text{ADIMENSIONALE e err. relativo}$$

ES:

$$P_1 = 1 \text{ bon} = 10^5 \text{ Pa}$$

$$P_1^* = 0,9 \text{ bon} = 0,9 \cdot 10^5 \text{ Pa}$$

$$E_{\text{Rom}} = \frac{0,1}{1} = 0,1 E_{\text{RELATIVO}}$$

$$= 0,1 E_{\text{REL}}$$

$$1) |b_m - a_m| \overset{10^{-8}}{< \text{TOLL}} \quad \frac{|b_m - a_m|}{b_0 - a_0} < \text{TOLL} \quad \boxed{X_{\text{TOL}} = 10^{-8}} < \text{EPS}$$

$$2) |F(x_m) - 0| \overset{10^{-6}}{< \text{TOLL}} \quad \frac{|F(x_m)|}{F_{\text{RIFERIMENTO}}} < \text{TOLLERANZA} \quad \boxed{F_{\text{TOL}} = 10^{-6}} < \text{EPS}$$

$$F_{\text{RIFERIMENTO}} = \max \{ |f(a)|, |f(b)| \}$$

ES: $F(x) = x^2 - 2$	$F(x) = 1000(x^2 - 2)$
$a = 0$	$a = 0$
$b = 2$	$b = 2$
$F(a) = -2$	$F(a) = -2000$
$F(b) = 2$	$F(b) = 2000$
$E_{\text{ASS}} = E_{\text{REL}}$	OK E_{ASS}

CARDO $10^6 \rightarrow 6$ cifre

$$F_{\text{REF}} = 1$$

CITRADO 6 cifre

$$F_{\text{REF}} = 1000$$

$$F(x) = \left(\frac{x}{1000} \right)^2 - 2$$

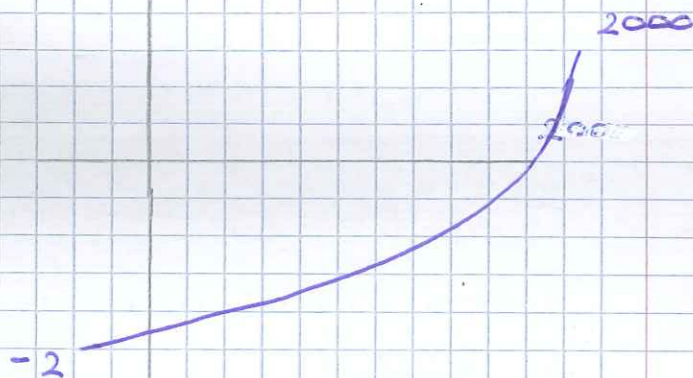
$$a = 0 \quad b = 2000 \quad (\text{ORDINE } 10^3)$$

$$F(a) = -2$$

$$F(b) = 2$$

$$E_A =$$

$$E_R =$$



PSEUDO-CODICE

ERRORI RELATIVI/ASSOLUTI NUMERICI

TEST ARRESTO

$$X_{\text{TOL}} = 10^{-8} > \text{EPS}$$

$$F_{\text{TOL}} = 10^{-6} > \text{EPS}$$

$$X_{\text{TOL}} = \max \{ \text{TOL}_x, 5\text{eps} \}$$

BISEZIONE (F, a, b, OPTIONS, TOLLERANZA X, TOLLERANZA Y, MAX ITERAZIONI, VERBOSE)

dei parametri α, β, γ

PER DEBUG
STAMPA ogni iterazione

$$F(x) = x^2 - \alpha$$

$$F(x, \alpha) = x^2 - \alpha$$

PARAMETRO

// PERCHÉ se si vuole fare la propria routine chiediamo che l'utente possa modificare soltanto la f

FACCIAMO in MATLAB UNA STRUTTURA

```

options = struct(); // creo una struct vuota e dinamicamente
                    // dichiaro i suoi campi: tolX, tolF
options.tolX = 1e-8; // TOLLERANZA X
options.tolF = 1e-6; // TOLLERANZA F
options.maxite = 1000;
options.verbose = TRUE;
    
```

se VERBOSE PER DICHIARARE UNA STRUCT in MATLAB

```
hist = struct(); // creo campo history
```

```
hist.xmedio = [a+b]/2
```

```
while ...
```

```
hist.xmedio = [hist.xmedio; xmedio] // per contemporaneamente ogni volta il nuovo xmedio
```

oppure per risparmiare spazio della memoria

```
hist.xmedio = repmat(NaN, 1, maxite) (NaN, NaN, NaN, NaN, NaN)
```

```
while ...
```

valore che non ha senso

```
hist.xmedio(i) = xmedio // sostituisco ogni volta il NaN il valore vero di xmedio
```

```
end;
```

```
hist.xmedio = hist.xmedio(1:i)
```

(NaN, NaN, NaN, NaN, NaN)
x_m, x_m, x_m, x_m, x_m

help newton serve per aggiungere dinamicamente i parametri della funzione di funzioni non lineari

Metodi per trovare le radici

METODO della TANGENTE

TANG
 → NEWTON
 → SECANTI
 ↘ TANGENTI FISSE

$$m = f'(x_m)$$

ATTRAVERSO ES della retta passante per un punto $y - y_A = m(x - x_A)$

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

ATTRAVERSO RETTA PASSANTE PER 2 punti

$$\frac{x - x_A}{x_B - x_A} = \frac{y - y_A}{y_B - y_A}$$

$$m = f'(x_c)$$

$(x_k, f(x_k))$

Equazione della retta passante per i punti $(x_k, f(x_k))$

IN questo modo CALCOLO IL punto SULL'ASSE X

$$\begin{cases} y - f(x_k) = m(x - x_k) \\ y = 0 \end{cases}$$

$$\Rightarrow 0 - f(x_k) = m(x_{k+1} - x_k)$$

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

ITERATA del metodo tangenti

TANGENTI FISSE

$$x_{m+1} = x_m - \frac{f(x_m)}{m}$$

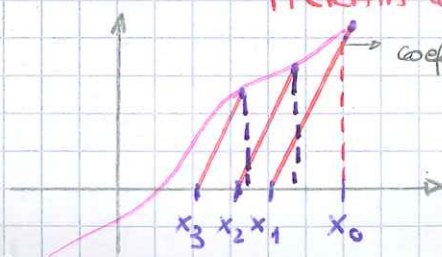
$m \rightarrow$ fissata (non cambiapendente)

NEWTON (TANGENTI VARIABILI)

$x_0 = a$

$x_0 = b$

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}$$



METODO SECANTI

L'idea è di approssimare $f'(x_m)$ che appare nel metodo di Newton con

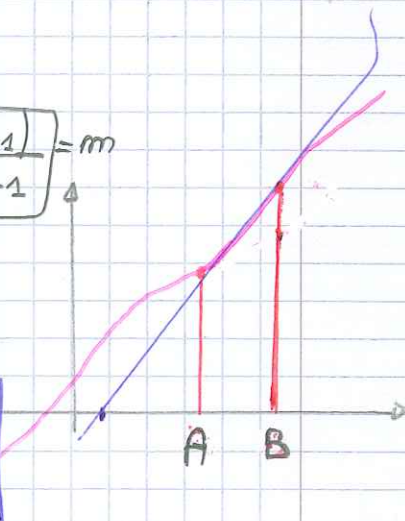
$$f'(x_m) = \frac{f(x_m) - f(x_{m-1})}{x_m - x_{m-1}} = m$$

$$x_{m+1} = x_m - \frac{f(x_m) \cdot (x_m - x_{m-1})}{f(x_m) - f(x_{m-1})}$$

ho bisogno di 2 punti retta passante per 2 punti

$$\begin{cases} x_0 = a \\ x_{m+1} = x_m - \frac{a - x_m}{f(a) - f(x_m)} \cdot f(x_m) \end{cases}$$

$$\begin{cases} x_0 = b \\ x_{m+1} = x_m - \frac{b - x_m}{f(b) - f(x_m)} \cdot f(x_m) \end{cases}$$



Regola generale

$f'(x) \cdot f(a) < 0$ pseudo $x_0 = a$
 $f'(x) \cdot f(b) > 0$ pseudo $x_0 = b$

ma punto da B

VALIDA per METODO della SECANTI

ma punto da A

SVILUPPO
TEYLOR

I^a ORDINE $\equiv$ LINEARIZZARE PROBLEMA

$$x_{m+1} \leftarrow x_m$$

$$f(x_{m+1}) = f(x_m + \delta)$$

$$= f(x_m) + f'(x_m) \cdot \delta + \overset{\text{ORDINE}}{O(\delta^2)}$$

$$0 = f(x_m) + f'(x_m) \cdot (x_{m+1} - x_m)$$

$$f(y) = f(x) + f'(x)(y-x) + \overset{\text{ORDINE}}{O((y-x)^2)}$$

$$\frac{f(y) - f(x)}{y - x} \approx f'(x) + \overset{\text{ORDINE}}{O(y-x)}$$

Es: Sviluppo in serie di TAYLOR

$$f(x) = e^x$$

$$e^x \approx 1 + x + \frac{x^2}{2!} + \dots$$

$$f(0) = e^0 = 1$$

$$f'(0) = 1$$

$$f(0)$$

$$f(x) = \cos(x)$$

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!}$$

Concludiamo ordine di convergenza dello sviluppo Newton

$$\lim_{m \rightarrow \infty} \frac{|\epsilon_{m+1}|}{|\epsilon_m|^p} = C \begin{matrix} \neq 0 \\ \neq \infty \end{matrix}$$

ϵ_{m+1} = errore di passo $m+1$
 ϵ_m = errore di passo m

$p=2$ ordine di convergenza quadratico

Con poche iterazioni (5/6) riusciamo a raggiungere tutte le cifre di precisione macchina (al massimo)

Metodo di Newton molto veloce per convergere però dobbiamo conoscere la derivata prima

DIMOSTRAZIONE VELOCITÀ di convergenza QUADRATICA

questo è vero se x_{m+1} vicino al valore della radice α (grazie soluzione)

perché se prendiamo un x_{m+1} lontano allora il metodo potrebbe divergere, oppure convergere lentamente

- ATTENZIONE: il metodo di Newton può divergere, invece il metodo di Bisezione converge sempre

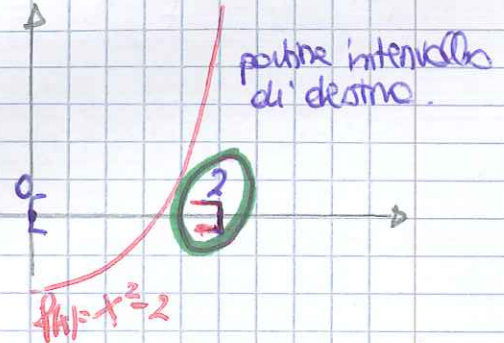
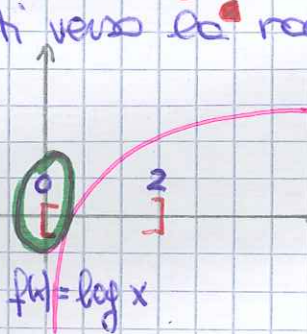
Quindi importante scegliere punto iniziale x_0

→ in questo modo il metodo risulta superefficiente.

- Come scegliere punto x_0 ? [La scelta avviene per visualizzazione grafica]

idea: La tangente punti verso la radice

Meglio partire da 0
intervallo di sinistra



scegliere sempre dalla parte della curva

Meglio anzi partire da 2 determinanti partire da 0 ma punta fuori strada

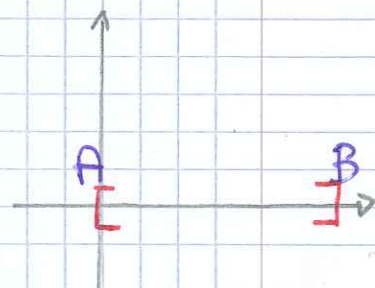
VALE SCONTRO PER METODO NEWTON

Partire dall'intervallo dx o sx dipende dalla derivata prima e seconda

REGOLA MONOTONICA

se $[f' \cdot f'' > 0]$ scelgo B altrimenti scelgo A

$f' \cdot f''$	$>$	$<$
$>$	B	A
$<$	A	B



oppure meglio quando

$f''(x) \cdot f(a) > 0$ $x_0 = a$ punto da A
 $f''(x) \cdot f(a) < 0$ $x_0 = b$ punto da A
 ed esso non è zero

Esempio: trova radice funzione $f(x) = x^2 - 2$ $f'(x) = 2x$

$$x_0 = ? \quad \boxed{x_0 = 2}$$

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

$$= 2 - \frac{(4-2)}{4} = 1,5$$

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

$$= 1,5 - \frac{(1,5)^2 - 2}{3} = \boxed{1,426} \quad \text{già raggiunto 2 cifre} \quad \approx \sqrt{2} = 1,4142$$

$$C = \frac{1}{2} \frac{f''(2)}{f'(2)} = \frac{1}{2} \frac{2}{2\sqrt{2}} \Rightarrow \boxed{C = \frac{\sqrt{2}}{4}} \approx 0,35$$

$$\frac{\text{Errore}_{m+1}}{\text{Errore}_m^2} \approx \boxed{0,35} \rightarrow \text{costante } C \text{ molto bene quindi errore prossimo allo zero}$$

Metodo di Newton Non funziona bene quando con errore ∞
 quando $f'(x) = 0$ ^{problema} la derivata prima = al valore della funzione in quel punto
 → Taylor perde ordini di convergenza

Esempio di malfunzionamento: FAIL TEST
 con $f(x) = x^2$

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)}$$

$$x_m - \frac{x_m^2}{2x_m} = \frac{x_m}{2} \quad x = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}$$

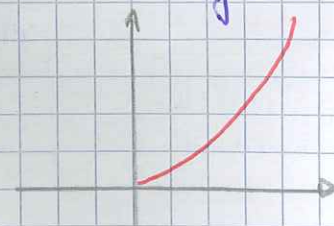
La generica iterata è $x_{m+1} = \frac{1}{2^{m+1}}$

$$\text{Errore al passo } m+1 = \left| 0 - \frac{1}{2^{m+1}} \right| = \frac{1}{2^{m+1}}$$

$$\text{Errore al passo } m = \frac{1}{2^m}$$

$$\frac{\text{Errore}_{m+1}}{\text{Errore}_m} = \frac{1}{2^m} \approx 2$$

però $p=2$
 questo metodo 1° ordine
 cioè converge linearmente



Metodo di Newton per

$$f(x) = x^2 - 9$$

$$\text{Radici } x = \pm\sqrt{9}$$

$$f'(x) = 2x$$

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

se applico al presente caso:

$$x_{k+1} = x_k - \frac{(x_k^2 - 9)}{2x_k}$$

$$= x_k - \frac{x_k}{2} + \frac{9}{2x_k}$$

$$= \frac{x_k}{2} + \frac{9}{2x_k}$$

$$x_{k+1} = \frac{1}{2} \left(x_k + \frac{9}{x_k} \right)$$

per calcolare la radice di un numero
basta 2 prodotti e 1 somma

Per trovare il reciproco di un numero

$$f(x) = \frac{1}{x} - 9$$

$$x^* = \frac{1}{9}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \Rightarrow x_k - \frac{\frac{1}{x_k} - 9}{-1/x_k^2} = x_k + \left(\frac{1}{x_k} - 9 \right) x_k^2$$
$$= 2x_k - 9x_k^2$$
$$= x_k(2 - 9x_k)$$

quindi per calcolare
il reciproco di un numero
basta 2 prodotti e 1 differenza

	ORDINE p
TANG FISSA	1
SECANTI	$1,618$ $1 < p < 2$
NEWTON	2

Iterazioni di punto fisso

$$x_{k+1} = y(x_k)$$

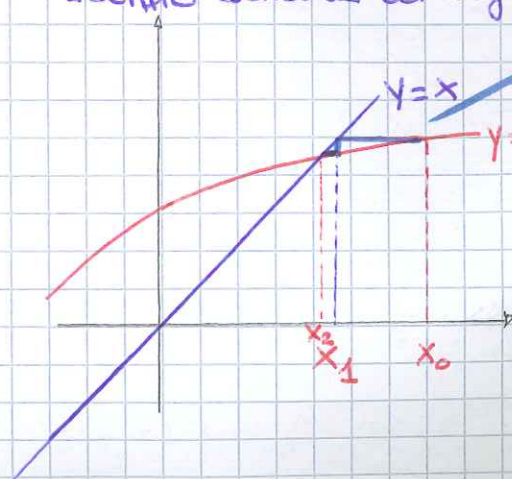
$$x = y(x)$$

$$F(x) = x^2 - 4$$

$$x_{k+1} = y(x_k)$$

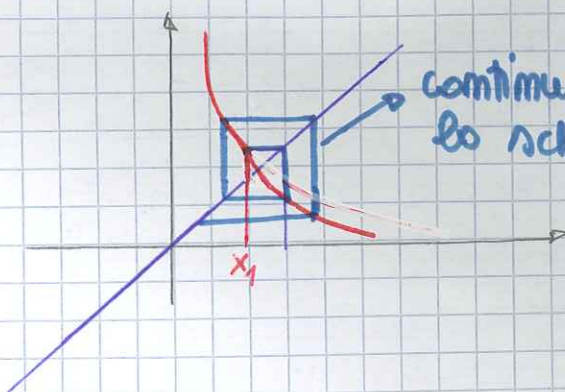
$$\begin{cases} y = x_{k+1} \\ y = g(x_k) \end{cases}$$

ESEMPIO schema converge



continuando ad iterare
la mia ALGORITMO
converge trova
il punto di
intersezione

ESEMPIO schema diverge



continuo ad iterare
lo schema diverge

$$1) x^{k+1} = (x^4 + x)^2$$

$$2) x^{k+1} = \frac{(4 + 11x - x^4)}{11}$$

$$F(x) = x^4 - 4$$

	$x^0 = 1$	x_1	x_2	x_3	x_4	
①	1	-2	10	10000	∞	DIVERGE
②	1	1,3	1,403	1,41		← CONVERGE

CALCOLO ERRORE per studiare la convergenza

SOLUZIONE x^*

$$x^* = g(x^*)$$

$$x^* - x^{k+1} = x^k - g(x^k) = g(x^*) - g(x^k) = g'(\xi_k)(x^* - x^k)$$

$$E_{k+1} = g'(\xi_k) E_k \leftarrow \text{da qui posso capire se la mia iterazione converge oppure no}$$

ITERO K volte

$$E_{k+1} = g'(\xi_k) E_k$$

$$E_k = g'(\xi_{k+1}) E_{k-1}$$

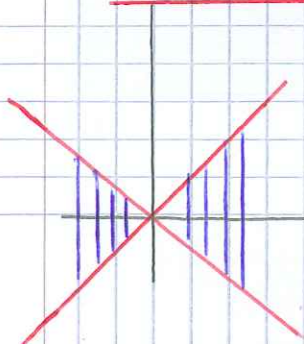
$$E_{k+1} = \underbrace{g'(\xi_k)}_{< m} + \underbrace{g'(\xi_{k-1})}_m \cdot E_{k-1}$$

$$E_{k+1} = m^k E_0$$

$m < 1$ cioè retta con coeff. angolare < 1

Il metodo CONVERGE quando $|g'(x)| < 1 \forall x \in I$

Se si verifica questa condizione il metodo CONVERGE



quando sto tra questo intervallo il coeff. angolare
ALLORA il metodo converge

Con quale ordine di convergenza il metodo converge?

$$x^{k+1} = g(x^k)$$

$$\underbrace{x^* - x^{k+1}}_{\substack{E_{k+1} \\ \text{sviluppo con Taylor}}} = f(x^*) - f(x^k) = f'(x^*)E_m + f''(x^*) \frac{E_m^2}{2!}$$

$$y(x^*) = f(x^k) + f'(x^k) \cdot (x^* - x^k) + f''(x^k) \cdot \frac{(x^* - x^k)^2}{2!} \dots$$

$$E_{m+1} = f'(x^k) \cdot E_m + f''(x^k) \frac{E_m^2}{2} + \dots$$

per calcolare l'ordine

$$\frac{E_{m+1}}{E_m^q} = C \begin{matrix} \neq 0 \\ \neq \infty \end{matrix}$$

- $x^{k+1} = g(x^k)$ se $f'(x^*) \neq 0$ LINEARE
- se $f'(x^*) = 0$ e $f''(x^*) \neq 0$ QUADRATICA

NEWTON fa $g(x) = x - \frac{f(x)}{f'(x)}$ iterazione punto fisso

$$g'(x) = 1 - \frac{f'(x) \cdot f(x) - f(x) \cdot f''(x)}{[f'(x)]^2} \overset{0}{=} 1 - 1 = 0$$

NEWTON ordine di convergenza

una funzione converge più velocemente rispetto ad un'altra tra g_1 (+ lenta), g_2 (+ veloce)

$$m_2 < m_1$$

più coeff. angolare piccolo più il metodo converge velocemente

L'iterazione di punto fisso ha un ordine di convergenza variabile dipende dalla forma della funzione, neutramente lineare