Equazioni non lineari: esempi.

- Risoluzione $f(x) = 0$ con $x \in [a, b] \subseteq \mathbb{R}$, $f \in C([a, b])$.

Equazioni non lineari: esempi.

- Risoluzione $f(x) = 0$ con $x \in [a, b] \subseteq \mathbb{R}$, $f \in C([a, b])$.
- Esempio 1: equazioni polinomiali $p_N(x) = 0$ con p_N polinomio di grado N. Possibili problemi (nessuna soluzione in $\mathbb{R}$, soluzioni multiple).

Equazioni non lineari: esempi.

- Risoluzione $f(x) = 0$ con $x \in [a, b] \subseteq \mathbb{R}$, $f \in C([a, b])$.
- Esempio 1: equazioni polinomiali $p_N(x) = 0$ con p_N polinomio di grado N. Possibili problemi (nessuna soluzione in $\mathbb{R}$, soluzioni multiple).
- Esempio 2: $f(x) = \sin(x) - x$. Soluzione unica poichè f decrescente (vedi derivata).

Equazioni non lineari: convergenza e ordine convergenza

- Metodo iterativo: genera successione $\{x_n\}_{n \in \mathbb{N}}$ che si desidera convergere a x^* tale che $f(x^*) = 0$.

Equazioni non lineari: convergenza e ordine convergenza

- Metodo iterativo: genera successione $\{x_n\}_{n\in\mathbb{N}}$ che si desidera convergere a x^* tale che $f(x^*) = 0$.
- **Ordine di convergenza** del metodo iterativo: sia $\{x_k\}$ una successione convergente ad x^* e sia $e_k = x_k - x^*$ l'errore al passo k. Se esiste un numero $p > 0$ e una costante $C \neq 0$ tale che

  $$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = C$$

  allora p è chiamato *ordine di convergenza* della successione e C è la *costante asintotica di errore*. Per $p = 1$ la convergenza di dice **lineare**, per $p = 2$ si dice **quadratica**.

Equazioni non lineari: esempio ordine di convergenza

Supponiamo

$$\frac{|e_{k+1}|}{|e_k|^p} = \frac{1}{10}, \ |e_0| = 1$$

o equivalentemente

$$|e_{k+1}| = \frac{1}{10} |e_k|^p, \ |e_0| = 1.$$

Il metodo ha ordine di convergenza p visto che

$$\lim_{n\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = \frac{1}{10},$$

- $p = 1$: $e_0 = 1$, $e_1 = 1/10$, $e_2 = 1/100$, $e_3 = 1/1000$, $\ldots$.

Equazioni non lineari: esempio ordine di convergenza

Supponiamo

$$\frac{|e_{k+1}|}{|e_k|^p} = \frac{1}{10}, \; |e_0| = 1$$

o equivalentemente

$$|e_{k+1}| = \frac{1}{10} |e_k|^p, \; |e_0| = 1.$$

Il metodo ha ordine di convergenza p visto che

$$\lim_{n \to \infty} \frac{|e_{k+1}|}{|e_k|^p} = \frac{1}{10},$$

- $p = 1$: $e_0 = 1$, $e_1 = 1/10$, $e_2 = 1/100$, $e_3 = 1/1000$, $\ldots$.
- $p = 2$: $e_0 = 1$, $e_1 = 1/10$, $e_2 = 1/1000$, $e_3 = 1/10^7, \ldots$.

Equazioni non lineari: esempio ordine di convergenza

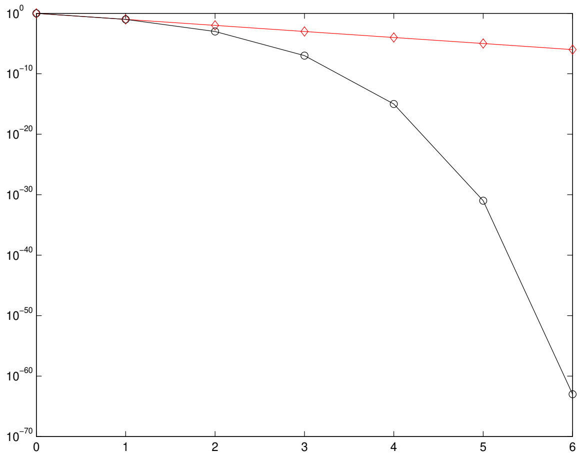

Figura: Grafico che illustra l'errore di un metodo con convergenza $p = 1$ (in rosso a rombi) e $p = 2$ (in nero a cerchietti), per $C = 1/10$ ed $e_0 = 1$.

Metodo bisezione: definizione

Sia $f : [a, b] \longrightarrow \mathbb{R}$ una funzione continua e supponiamo $f(a) \cdot f(b) < 0$. Il metodo di bisezione genera una successione di intervalli (a_k, b_k) con

- $f(a_k) \cdot f(b_k) < 0$,
- $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$,
- $|b_k - a_k| = \frac{1}{2}|b_{k-1} - a_{k-1}|$.

Fissate due tolleranze ϵ_1, ϵ_2 si arresta l'algoritmo quando

$$|b_k - a_k| \leq \epsilon_1 \text{ oppure } |f((a_k + b_k)/2)| \leq \epsilon_2.$$

Metodo bisezione: esempio

Operativamente dati $a \leq b$ con $f(a) \cdot f(b) < 0$ calcola $c = (a+b)/2$. Se $f(a) \cdot f(c) > 0$ sostituisce c ad a, viceversa sostituisce c a b, fermandosi se le condizioni d'arresto sono verificate.

Studiamo $f(x) = 0$ con $f(x) = \sin(x) - x$. Osserviamo che se $x < 0$ allora $f(x) > 0$ altrimenti $f(x) \leq 0$.

1. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.40000000 \rightarrow \mathbf{c} = 0.05000000$.

Metodo bisezione: esempio

Operativamente dati $a \leq b$ con $f(a) \cdot f(b) < 0$ calcola $c = (a+b)/2$. Se $f(a) \cdot f(c) > 0$ sostituisce c ad a, viceversa sostituisce c a b, fermandosi se le condizioni d'arresto sono verificate.

Studiamo $f(x) = 0$ con $f(x) = \sin(x) - x$. Osserviamo che se $x < 0$ allora $f(x) > 0$ altrimenti $f(x) \leq 0$.

1. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.40000000 \rightarrow \mathbf{c} = 0.05000000$.
2. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.12500000$

Metodo bisezione: esempio

Operativamente dati $a \leq b$ con $f(a) \cdot f(b) < 0$ calcola $c = (a+b)/2$. Se $f(a) \cdot f(c) > 0$ sostituisce c ad a, viceversa sostituisce c a b, fermandosi se le condizioni d'arresto sono verificate.

Studiamo $f(x) = 0$ con $f(x) = \sin(x) - x$. Osserviamo che se $x < 0$ allora $f(x) > 0$ altrimenti $f(x) \leq 0$.

1. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.40000000 \rightarrow \mathbf{c} = 0.05000000$.
2. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.12500000$
3. $\mathbf{a} = -0.12500000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.03750000$

Metodo bisezione: esempio

Operativamente dati $a \leq b$ con $f(a) \cdot f(b) < 0$ calcola $c = (a+b)/2$. Se $f(a) \cdot f(c) > 0$ sostituisce c ad a, viceversa sostituisce c a b, fermandosi se le condizioni d'arresto sono verificate.

Studiamo $f(x) = 0$ con $f(x) = \sin(x) - x$. Osserviamo che se $x < 0$ allora $f(x) > 0$ altrimenti $f(x) \leq 0$.

1. $\mathbf{a} = -0.30000000,\ \mathbf{b} = 0.40000000 \rightarrow \mathbf{c} = 0.05000000.$
2. $\mathbf{a} = -0.30000000,\ \mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.12500000$
3. $\mathbf{a} = -0.12500000,\ \mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.03750000$
4. $\mathbf{a} = -0.03750000,\ \mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = 0.00625000$

Metodo bisezione: esempio

Operativamente dati $a \leq b$ con $f(a) \cdot f(b) < 0$ calcola $c = (a+b)/2$. Se $f(a) \cdot f(c) > 0$ sostituisce c ad a, viceversa sostituisce c a b, fermandosi se le condizioni d'arresto sono verificate.

Studiamo $f(x) = 0$ con $f(x) = \sin(x) - x$. Osserviamo che se $x < 0$ allora $f(x) > 0$ altrimenti $f(x) \leq 0$.

1. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.40000000 \rightarrow \mathbf{c} = 0.05000000$.
2. $\mathbf{a} = -0.30000000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.12500000$
3. $\mathbf{a} = -0.12500000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = -0.03750000$
4. $\mathbf{a} = -0.03750000$, $\mathbf{b} = 0.05000000 \rightarrow \mathbf{c} = 0.00625000$
5. ...

Metodo bisezione: alcuni fatti

- Non ha una velocità di convergenza nel senso della definizione sopra definita. Esempio: se applico bisezione per risolvere $\sin(x) - x = 0$, in $[a, b] = [-3, 2]$ la successione $|e_{n+1}/e_n|$ alterna valori 1.5 e 1/6 e quindi non converge con ordine 1.

Metodo bisezione: alcuni fatti

- Non ha una velocità di convergenza nel senso della definizione sopra definita. Esempio: se applico bisezione per risolvere $\sin(x) - x = 0$, in $[a, b] = [-3, 2]$ la successione $|e_{n+1}/e_n|$ alterna valori 1.5 e 1/6 e quindi non converge con ordine 1.
- Usa solo il segno della funzione.

Metodo bisezione: alcuni fatti

- Non ha una velocità di convergenza nel senso della definizione sopra definita. Esempio: se applico bisezione per risolvere $\sin(x) - x = 0$, in $[a, b] = [-3, 2]$ la successione $|e_{n+1}/e_n|$ alterna valori 1.5 e 1/6 e quindi non converge con ordine 1.
- Usa solo il segno della funzione.
- Se $f(a) \cdot f(b) < 0$ e $f \in C([a, b])$ allora converge (sempre!!) a un x^* tale che $f(x^*) = 0$.

Metodo bisezione: alcuni fatti

- Non ha una velocità di convergenza nel senso della definizione sopra definita. Esempio: se applico bisezione per risolvere $\sin(x) - x = 0$, in $[a, b] = [-3, 2]$ la successione $|e_{n+1}/e_n|$ alterna valori 1.5 e 1/6 e quindi non converge con ordine 1.
- Usa solo il segno della funzione.
- Se $f(a) \cdot f(b) < 0$ e $f \in C([a, b])$ allora converge (sempre!!) a un x^* tale che $f(x^*) = 0$.
- Il test del residuo può essere non adatto a funzioni *piatte* o con *picchi* intorno al punto cui converge.

Metodo bisezione: alcuni fatti

- Non ha una velocità di convergenza nel senso della definizione sopra definita. Esempio: se applico bisezione per risolvere $\sin(x) - x = 0$, in $[a, b] = [-3, 2]$ la successione $|e_{n+1}/e_n|$ alterna valori 1.5 e 1/6 e quindi non converge con ordine 1.
- Usa solo il segno della funzione.
- Se $f(a) \cdot f(b) < 0$ e $f \in C([a, b])$ allora converge (sempre!!) a un x^* tale che $f(x^*) = 0$.
- Il test del residuo può essere non adatto a funzioni *piatte* o con *picchi* intorno al punto cui converge.
- Fissata una tolleranza ϵ, e due punti iniziali a, b tali che $f(a) \cdot f(b) < 0$ (con $f \in C([a, b])$) per avere un'errore assoluto $|x_n - x^*|$ sulla soluzione x^* inferiore ad ϵ necessitano al piú

  $$n = \texttt{ceil}\left(\frac{\log(b-a) - \log\epsilon}{\log 2}\right)$$

  iterazioni del metodo.

Metodo Newton: interpretazione analitica

Il metodo di Newton richiede che

- f sia derivabile con continuità su un intervallo $[a, b]$ di $\mathbb{R}$;

Se $x_k \in [a, b]$ è l'ultima approssimazione del metodo, allora dalla formula di Taylor centrata in x_k abbiamo per $\xi_k \in \mathcal{I}(x^*, x_k)$

$$0 = f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + f''(\xi_k)(x^* - x_k)^2/2$$

Tralasciando i termini di ordine superiore

$$0 \approx f(x_k) + f'(x_k)(x^* - x_k)$$

e quindi se $f^{(1)}(x_k) \neq 0$, dopo facili conti

$$x^* \approx x_{k+1} := x_k - f(x_k)/f^{(1)}(x_k).$$

Metodo Newton: interpretazione analitica

Il metodo di Newton richiede che

- f sia derivabile con continuità su un intervallo $[a, b]$ di $\mathbb{R}$;
- $f^{(1)}(x) \neq 0$ in $[a, b]$.

Se $x_k \in [a, b]$ è l'ultima approssimazione del metodo, allora dalla formula di Taylor centrata in x_k abbiamo per $\xi_k \in \mathcal{I}(x^*, x_k)$

$$0 = f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + f''(\xi_k)(x^* - x_k)^2/2$$

Tralasciando i termini di ordine superiore

$$0 \approx f(x_k) + f'(x_k)(x^* - x_k)$$

e quindi se $f^{(1)}(x_k) \neq 0$, dopo facili conti

$$x^* \approx x_{k+1} := x_k - f(x_k)/f^{(1)}(x_k).$$

Metodo Newton: interpretazione geometrica

Se ben definito, il metodo di Newton genera una successione $\{x_k\}$ definita da

$$x_{k+1} := x_k - f(x_k)/f^{(1)}(x_k).$$

Sia **r** la retta tangente al grafico della funzione f nel punto $(x_k, f(x_k))$. Allora x_{k+1} è l'intersezione della retta **r** con l'asse delle ascisse, cioè la retta di equazione $y = 0$.

Metodo Newton: interpretazione geometrica

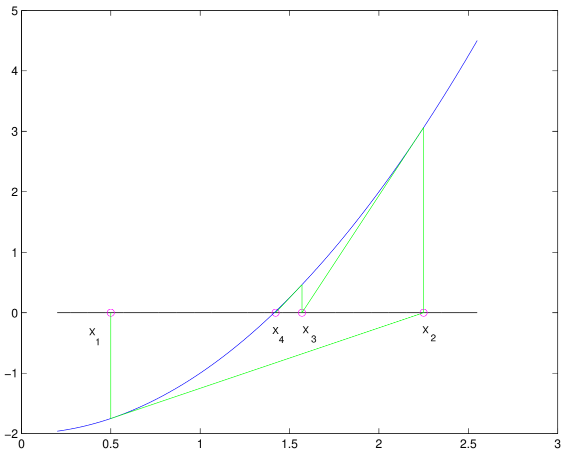

Figura: Grafico che illustra geometricamente le iterazioni del metodo Newton per il calcolo dello zero di $f(x) = x^2 - 2$.

Metodo Newton: convergenza

Nel caso del metodo di Newton la convergenza non è in generale garantita. Esistono degli esempi in cui il metodo produce una successione non convergente. Ciò nonostante esistono molti teoremi che illustrano quando si è certi che il metodo converga.

- Alcuni sono detti di convergenza locale e dicono che se x_0 è in un intorno $\mathcal{I}$ sufficientemente piccolo della soluzione x^* allora il metodo converge ad x_*. Usualmente non sono costruttivi e non permettono di definire chiaramente $\mathcal{I}$.

Metodo Newton: convergenza

Nel caso del metodo di Newton la convergenza non è in generale garantita. Esistono degli esempi in cui il metodo produce una successione non convergente. Ciò nonostante esistono molti teoremi che illustrano quando si è certi che il metodo converga.

- Alcuni sono detti di convergenza locale e dicono che se x_0 è in un intorno $\mathcal{I}$ sufficientemente piccolo della soluzione x^* allora il metodo converge ad x_*. Usualmente non sono costruttivi e non permettono di definire chiaramente $\mathcal{I}$.
- Altri sono detti di convergenza globale e dicono che se x_0 appartiene a un ben definito intorno $\mathcal{I}$ di x^* allora il metodo converge.

Metodo Newton: un teorema di convergenza locale

Uno zero x^* si dice semplice se $f(x^*) = 0$ e $f^{(1)}(x^*) \neq 0$.

Teorema. Sia $x^* \in (a, b)$ uno zero semplice di $f : [a, b] \to \mathbb{R}$. Si supponga inoltre $f \in C^2([a, b])$. Allora per $x_0 \in [a, b]$ sufficientemente vicino a x^* le iterazioni del metodo di Newton

$$x_{k+1} = x_k - f(x_k)/f^{(1)}(x_k), \;\; k = 0, 1, 2, \ldots$$

sono ben definite e convergono quadraticamente a x^*.

Metodo Newton: un teorema di convergenza locale (dim.)

Traccia della dimostrazione.

Sia I_1 intorno di x^* in cui $f^{(1)}$ ha segno costante (permanenza del segno).

1. Dalla formula di Taylor centrata in x^* e valutata in x_k abbiamo per $\xi_k \in \mathcal{I}(x^*, \xi_k)$, ricordando la succ. del metodo di Newton

$$\begin{aligned} 0 &= f(x^*) = f(x_k) + f^{(1)}(x_k)(x^* - x_k) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \\ &= f^{(1)}(x_k)(x^* - x_{k+1}) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \qquad (1) \end{aligned}$$

Posto $e_k = |x^* - x_k|$ ricaviamo per $M = \max_{x,y \in I_1} |f^{(2)}(\xi_k)|/2 \cdot |f^{(1)}(x_k)|$

$$e_{k+1} = \frac{|f^{(2)}(\xi_k)|}{2 \cdot |f^{(1)}(x_k)|} e_k^2 \leq M {e_k}^2$$

Metodo Newton: un teorema di convergenza locale (dim.)

Sia $I_2 = [x^* - \delta, x^* + \delta] \subseteq I_1$ con $M\delta < 1$. Se $x_0 \in I_2$ allora $e_0 \leq \delta$ e

$$e_1 \leq M{e_0}^2 \leq M\delta^2 \leq \delta$$

da cui per induzione ogni $x_k \in I_2$. Inoltre

$$Me_k \leq M^2{e_{k-1}}^2 = (Me_{k-1})^2 \leq \ldots \leq (Me_0)^{2^k}$$

e visto che $Me_0 \leq M\delta < 1$ abbiamo $e_k \to 0$, cioè il metodo di Newton converge quadraticamente. Infatti si vede facilmente che essendo $\xi_k \in \mathcal{I}(x^*, x_k)$ e $x_k \to x^*$ necessariamente $f \in C^{(2)}([a, b])$

$$\begin{aligned} \lim_k \frac{e_{k+1}}{e_k^2} &= \lim_k |f^{(2)}(\xi_k)|/(2 \cdot |f^{(1)}(x_k)|) \\ &= \lim_k |f^{(2)}(x^*)|/(2 \cdot |f^{(1)}(x^*)|) \neq 0 \end{aligned}$$

Metodo Newton: un teorema di convergenza

Teorema. Sia $f \in C^2([a, b])$, con $[a, b]$ intervallo chiuso e limitato. Se

1. $f(a) \cdot f(b) < 0$;
2. $f^{(1)}(x) > 0$, per ogni $x \in [a, b]$;
3. $f^{(2)}(x) > 0$

allora le iterate x_k fornite dal metodo di Newton sono strettamente decrescenti e convergono all'unica soluzione x^* in $[a, b]$, per $x_0 = b$.

Metodo Newton: un teorema di convergenza globale

Teorema.

Sia $f \in C^2([a, b])$, con $[a, b]$ intervallo chiuso e limitato. Se

1. $f(a) \cdot f(b) < 0$;
2. $f^{(1)}(x) \neq 0$, per ogni $x \in [a, b]$;
3. $f^{(2)}(x) \geq 0$ o $f^{(2)}(x) \leq 0$, per ogni $x \in [a, b]$;
4. $|f(a)/f^{(1)}(a)| < b - a$ e $|f(b)/f^{(1)}(b)| < b - a$,

allora il metodo di Newton converge all'unica soluzione x^* in $[a, b]$, per ogni $x_0 \in [a, b]$.

Metodo Newton: un teorema di convergenza globale (dim.)

Traccia della dimostrazione.

Supponiamo che

1. $f^{(2)}(x) \geq 0$ se $x \in (a, b)$.
2. $f^{(1)}(x) > 0$ se $x \in (a, b)$.

Allora:

1. La soluzione x^* è unica poichè f cambia segno ed è crescente.

Metodo Newton: un teorema di convergenza globale (dim.)

Traccia della dimostrazione.

Supponiamo che

1. $f^{(2)}(x) \geq 0$ se $x \in (a, b)$.
2. $f^{(1)}(x) > 0$ se $x \in (a, b)$.

Allora:

1. La soluzione x^* è unica poichè f cambia segno ed è crescente.
2. L'ultimo punto garantisce che $x_1 \in (a, b)$.

Metodo Newton: un teorema di convergenza globale (dim.)

Traccia della dimostrazione.

Supponiamo che

1. $f^{(2)}(x) \geq 0$ se $x \in (a, b)$.
2. $f^{(1)}(x) > 0$ se $x \in (a, b)$.

Allora:

1. La soluzione x^* è unica poichè f cambia segno ed è crescente.
2. L'ultimo punto garantisce che $x_1 \in (a, b)$.
3. Da hp. 2 e $f(x_k) + f^{(1)}(x_k)(x^* - x_k) = f^{(1)}(x_k)(x^* - x_{k+1})$:

$$\begin{aligned} 0 &= f(x^*) = f(x_k) + f^{(1)}(x_k)(x^* - x_k) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \\ &= f^{(1)}(x_k)(x^* - x_{k+1}) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \end{aligned}$$

e quindi $0 \geq f^{(1)}(x_k)(x^* - x_{k+1})$ cioè $x^* \leq x_{k+1}$.

Metodo Newton: un teorema di convergenza globale (dim.)

Traccia della dimostrazione.

Supponiamo che

1. $f^{(2)}(x) \geq 0$ se $x \in (a,b)$.
2. $f^{(1)}(x) > 0$ se $x \in (a,b)$.

Allora:

1. La soluzione x^* è unica poichè f cambia segno ed è crescente.
2. L'ultimo punto garantisce che $x_1 \in (a,b)$.
3. Da hp. 2 e $f(x_k) + f^{(1)}(x_k)(x^* - x_k) = f^{(1)}(x_k)(x^* - x_{k+1})$:

$$\begin{aligned} 0 &= f(x^*) = f(x_k) + f^{(1)}(x_k)(x^* - x_k) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \\ &= f^{(1)}(x_k)(x^* - x_{k+1}) + f^{(2)}(\xi_k)(x^* - x_k)^2/2 \end{aligned}$$

   e quindi $0 \geq f^{(1)}(x_k)(x^* - x_{k+1})$ cioè $x^* \leq x_{k+1}$.
4. La formula di Newton $x_{k+1} = x_k - f(x_k)/f^{(1)}(x_k)$ mostra che se $x_k > x^*$ allora $x_{k+1} < x_k$ (cioè $\{x_k\}_{k=1,\ldots}$ decrescente).

Metodo Newton: un teorema di convergenza globale (dim.)

Quindi, comunque sia scelto $x_0 \in [a, b]$, abbiamo che per $k \geq 1$ si ha $x_k \in [x^*, b]$ e $\{x_k\}_{k=1,\ldots}$ decrescente e quindi converge a un certo $L \in [x^*, b]$. Dalla formula del metodo di Newton, per continuità

$$\begin{aligned} L &= \lim_k x_{k+1} = \lim_k (x_k - f(x_k)/f^{(1)}(x_k)) \\ &= \lim_k x_k - \lim_k f(x_k)/f^{(1)}(x_k) \\ &= \lim_k x_k - \lim_k f(x_k)/\lim_k f^{(1)}(x_k) = L - f(L)/f^{(1)}(L) \end{aligned}$$

da cui $F(L)/f^{(1)}(L) = 0$ ed essendo $f^{(1)}(x) > 0$ per $x \in [a, b]$ abbiamo $F(L) = 0$. Ma x^* è l'unico zero di f per cui $L = x^*$ e $x_k \rightarrow x^*$.

Metodo Newton: alcuni fatti

1. Il metodo di Newton non è sempre convergente.

Metodo Newton: alcuni fatti

1. Il metodo di Newton non è sempre convergente.
2. Se converge, non è detto che l'ordine di convergenza sia $p = 2$ (conv. quadratica).

Metodo Newton: alcuni fatti

1. Il metodo di Newton non è sempre convergente.
2. Se converge, non è detto che l'ordine di convergenza sia $p = 2$ (conv. quadratica).
3. Se uno zero x^* di f non è semplice allora la convergenza non è quadratica.

Metodi di punto fisso

Consideriamo i problemi di punto fisso

$$x = \phi(x)$$

in cui supponiamo che $\phi : \Omega \rightarrow \Omega$ sia una funzione continua in Ω sia un intervallo compatto di $\mathbb{R}$. Notiamo subito che ogni problema di tipo $f(x) = 0$ si può ovviamente riscrivere in questa forma, posto $\phi(x) = f(x) + x$. Nel metodo di punto fisso, si definisce la successione

$$x_{k+1} = \phi(x_k)$$

Teorema di punto fisso

Teorema. Si supponga che $\phi : \Omega \rightarrow \Omega$ e ϕ' siano funzione continue in $\Omega = [a, b]$ (intv. compatto). Inoltre sia $\lambda = \max_{x \in \Omega} |\phi'(x)| < 1$. Allora:

1. Esiste uno ed un solo α tale che $\phi(\alpha) = \alpha$ in Ω;

Teorema di punto fisso

Teorema. Si supponga che $\phi : \Omega \to \Omega$ e ϕ' siano funzione continue in $\Omega = [a, b]$ (intv. compatto). Inoltre sia $\lambda = \max_{x \in \Omega} |\phi'(x)| < 1$. Allora:

1. Esiste uno ed un solo α tale che $\phi(\alpha) = \alpha$ in Ω;
2. Per ogni punto iniziale $x_0 \in \Omega$ le iterate $x_{n+1} = \phi(x_n)$ convergono ad α;

Teorema di punto fisso

Teorema. Si supponga che $\phi : \Omega \to \Omega$ e ϕ' siano funzione continue in $\Omega = [a, b]$ (intv. compatto). Inoltre sia $\lambda = \max_{x \in \Omega} |\phi'(x)| < 1$. Allora:

1. Esiste uno ed un solo α tale che $\phi(\alpha) = \alpha$ in Ω;
2. Per ogni punto iniziale $x_0 \in \Omega$ le iterate $x_{n+1} = \phi(x_n)$ convergono ad α;
3. Si ha

$$|\alpha - x_n| \leq \frac{\lambda^n}{1 - \lambda} |x_0 - x_1|, \;\; n \geq 0;$$

Teorema di punto fisso

Teorema. Si supponga che $\phi : \Omega \to \Omega$ e ϕ' siano funzione continue in $\Omega = [a, b]$ (intv. compatto). Inoltre sia $\lambda = \max_{x \in \Omega} |\phi'(x)| < 1$. Allora:

1. Esiste uno ed un solo α tale che $\phi(\alpha) = \alpha$ in Ω;
2. Per ogni punto iniziale $x_0 \in \Omega$ le iterate $x_{n+1} = \phi(x_n)$ convergono ad α;
3. Si ha
$$|\alpha - x_n| \leq \frac{\lambda^n}{1-\lambda}|x_0 - x_1|, \ n \geq 0;$$
4. Infine
$$\lim_{n \to \infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \phi'(\alpha)$$

Metodi di punto fisso

Consideriamo un generico problema di punto fisso $x = \phi(x)$, e supponiamo

1. ϕ di classe $C^{(2)}$,
2. $\phi(\alpha) = \alpha$,
3. $\phi'(\alpha) = 0$, $\phi''(\alpha) \neq 0$,
4. $x_n \rightarrow \alpha$.

Allora per $\xi_n \in \mathcal{I}(\alpha, x_n)$,

$$\begin{aligned} x_{n+1} = \phi(x_n) &= \phi(\alpha) + \phi'(\alpha)(x_n - \alpha) + \phi''(\xi_n)(x_n - \alpha)^2/2 \\ &= \alpha + \phi''(\xi_n)(x_n - \alpha)^2/2 \end{aligned}$$

e quindi facilmente portando α a primo membro, per $e_n = |x_n - \alpha|$

$$e_{n+1} = |\phi''(\xi_n)| e_n^2/2 \Rightarrow \lim_{n\rightarrow\infty} e_{n+1}/e_n^2 = \lim_k |\phi''(\xi_n)|/2 = |\phi''(\alpha)|/2$$

poichè $\xi_n \in \mathcal{I}(\alpha, x_n)$ e $x_n \rightarrow \alpha$, e $\phi'' \in C^2([a, b])$ per hp.1. Quindi la convergenza è quadratica.

Metodi di punto fisso e Newton

Il metodo di Newton è un particolare metodo di punto fisso. Infatti posto

$$\phi(x) = x - f(x)/f'(x)$$

abbiamo che $x_{k+1} = \phi(x_k) = x_k - f(x_k)/f'(x_k)$. Osserviamo che se $f(\alpha) = 0$ e $f'(\alpha) = 0$ allora

$$\phi(\alpha) = \alpha - f(\alpha)/f'(\alpha) = \alpha$$

cioè lo zero della f è pure sol. del problema di pto fisso e viceversa. Quindi se $f \in C^2(\Omega)$, $f'(x) \neq 0$, $\phi : \Omega \to \Omega$ otteniamo

$$\phi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$$

e se $f'(\alpha) \neq 0$ (radice semplice) abbiamo $\phi'(\alpha) = 0$ da cui deduciamo nuovamente che la convergenza è quadratica.

Cenni alla risoluzione numerica di sistemi nonlineari

Problema. Data una funzione $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ si tratta di trovare gli $x^* \in \Omega$ tali che $F(x^*) = 0$.

Esempio:

$$\left\{ \begin{array}{l} x - \frac{1}{2}\cos(y) = 0 \\ y - \frac{1}{2}\sin(x) = 0 \end{array} \right. \tag{2}$$

Soluzione (unica!):

$$x = 0.48640515466592, \;\; y = 0.23372550195872.$$

Generalizzazione punto fisso e Teorema di Banach

Teorema. Sia (X, d) uno spazio metrico completo ed M un sottoinsieme non vuoto e chiuso di X. Se $\phi : M \rightarrow M$ è L contrattiva cioè $d(\phi(\mathbf{x}), \phi(\mathbf{y})) \leq L d(\mathbf{x}, \mathbf{y})$ con $L < 1$ allora

1. l'equazione $\mathbf{x} = \phi(\mathbf{x})$ ha una ed una sola soluzione $\mathbf{x}^*$;
2. la successione $\mathbf{x_{k+1}} = \phi(\mathbf{x_k})$ (detta di Banach o di punto fisso) converge alla soluzione $\mathbf{x}^*$ per ogni scelta del punto iniziale $\mathbf{x_0}$;
3. per $k = 0, 1, 2, \ldots$ si ha
$$d(\mathbf{x_k}, \mathbf{x}^*) \leq L^k (1 - L)^{-1} d(\mathbf{x_0}, \mathbf{x_1}), \text{ a priori}$$
$$d(\mathbf{x_{k+1}}, \mathbf{x}^*) \leq L (1 - L)^{-1} d(\mathbf{x_{k+1}}, \mathbf{x_k}), \text{ a posteriori}$$
4. per $k = 0, 1, 2, \ldots$ si ha
$$d(\mathbf{x_{k+1}}, \mathbf{x}^*) \leq L d(\mathbf{x_k}, \mathbf{x}^*)$$

Punto fisso per sistemi e Teorema di Banach

Teorema. Sia $\mathbb{R}^d$ (dotato della norma euclidea $\|x\| = \sqrt{\sum_{k=1}^{d} x_k^d}$) ed M un sottoinsieme non vuoto e chiuso di $\mathbb{R}^d$. Se $\phi : M \rightarrow M$ è L contrattiva cioè $\|\phi(\mathbf{x}) - \phi(\mathbf{y}\| \leq L\|\mathbf{x} - \mathbf{y}\|$ con $L < 1$ allora

1. l'equazione $\mathbf{x} = \phi(\mathbf{x})$ ha una ed una sola soluzione $\mathbf{x}^*$;
2. la successione $\mathbf{x_{k+1}} = \phi(\mathbf{x_k})$ (detta di Banach o di punto fisso) converge alla soluzione $\mathbf{x}^*$ per ogni scelta del punto iniziale $\mathbf{x_0}$;
3. per $k = 0, 1, 2, \ldots$ si ha
$$\|\mathbf{x_k} - \mathbf{x}^*\| \leq L^k(1 - L)^{-1}\|\mathbf{x_0} - \mathbf{x_1}\|, \text{ a priori}$$
$$\|\mathbf{x_{k+1}} - \mathbf{x}^*\| \leq L(1 - L)^{-1}\|\mathbf{x_{k+1}} - \mathbf{x_k}\|, \text{ a posteriori}$$
4. per $k = 0, 1, 2, \ldots$ si ha
$$\|\mathbf{x_{k+1}} - \mathbf{x}^*\| \leq L\|\mathbf{x_k} - \mathbf{x}^*\|$$

Esempio punto fisso per sistemi

Esempio: il sistema (2) è eqv. a risolvere il problema di punto fisso

$$\left\{ \begin{array}{l} x = \frac{1}{2}\cos(y) \\ y = \frac{1}{2}\sin(x) \end{array} \right. \tag{3}$$

che si scrive come $\mathbf{v} = \phi(\mathbf{v})$ con $\mathbf{v} = (x, y)$ e

$$\phi(\mathbf{v}) = \left(\frac{1}{2}\cos(y), \frac{1}{2}\sin(x) \right).$$

Si dimostra che ϕ è L contrazione con $L = 1/2$. Per applicare il teorema di punto fisso per sistemi basta porre

$$M = [-1, 1] \times [-1, 1] \subset \mathbb{R}^2 \text{ (quadrato unitario)}$$

e osservare che

$$\phi(M) \subseteq [-1/2, 1/2] \times [-1/2, 1/2] \subset M$$

in quanto $-1 \leq \cos(x) \leq 1$, $-1 \leq \sin(x) \leq 1$.

L contrattiva (per esperti)

Mostriamo che $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ è contrattiva. Dalla Taylor in $\mathbb{R}^d$

$$\phi(\mathbf{v}) = \phi(\mathbf{v_0}) + (\phi'(\xi)) \cdot (\mathbf{v} - \mathbf{v_0}),\ \xi \in S$$

con S il segmento che congiunge $\mathbf{v}$ con $\mathbf{v_0}$, ϕ' la matrice Jacobiana. Nel nostro esempio quindi,

$$(\phi')_{j,k}(\mathbf{v}) = \begin{pmatrix} 0 & (-1/2)\sin(v_1) \\ (+1/2)\cos(v_2) & 0 \end{pmatrix}$$

Essendo $\|A\|_\infty = \max_j \sum_i |a_{i,j}|$ si ha

$$\begin{aligned} \|(\phi')(\mathbf{v})\|_\infty &= \max_j \sum_i |\left((\phi')(\mathbf{v})\right)_{i,j}| \\ &= \max\left(|(-1/2)\sin(v_1)|, |(+1/2)\cos(v_2)|\right) \leq \frac{1}{2} \end{aligned}$$

Quindi

$$\|\phi(\mathbf{v}) - \phi(\mathbf{v_0})\|_\infty \leq \|(\phi'(\xi))\|_\infty \|(\mathbf{v} - \mathbf{v_0})\|_\infty,\ \xi \in S$$

da cui $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ è una L-contrazione con $L = 1/2$.

Esempio punto fisso per sistemi

Successione punto fisso (convergente alla soluzione di (3)):

$$x_{k+1} = \frac{1}{2}\cos(y_k), \quad y_{k+1} = \frac{1}{2}\sin(x_k)$$

x_k	y_k
0.00000000000000	0.00000000000000
0.50000000000000	0.00000000000000
0.50000000000000	0.23971276930210
0.48570310513698	0.23971276930210
0.48570310513698	0.23341513183103
0.48644107261515	0.23341513183103
...	...
0.48640515466592	0.23372550195872

Metodo Newton per sistemi

Supponiamo di dover risolvere $f(\mathbf{v}) = \mathbf{0}$ con $f : \Omega \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^d$.

La successione di Newton viene descritta come

$$\begin{cases} f'(\mathbf{v}_k)) \cdot h_{k+1} = -f(\mathbf{v}_k), \\ \mathbf{v_{k+1}} := \mathbf{v_k} + h_{k+1} \end{cases} \tag{4}$$

dove f' è la matrice Jacobiana di f cioè se $\mathbf{v} = (v_1, \ldots, v_d)$ allora

$$(f'(\mathbf{v}_k)))_{i,j} = \frac{\partial f_i(\mathbf{v})}{\partial v_j}.$$

Si noti che ad ogni iterazione si deve risolvere il sistema lineare

$$f'(\mathbf{v}_k)) \cdot h_{k+1} = -f(\mathbf{v}_k).$$

Esempio Newton per sistemi

Consideriamo il precedente esempio, come $f(x) = 0$. Allora posto $\mathbf{v} = (x, y)$ si ha $f(x, y) = (f_1(x, y), f_2(x, y))$ con

$$\begin{aligned} f_1(x, y) &= x - \frac{1}{2}\cos(y) \\ f_2(x, y) &= y - \frac{1}{2}\sin(x). \end{aligned}$$

Inoltre

$$f'(x, y) = \begin{pmatrix} 1 & \frac{1}{2}\sin(y) \\ -\frac{1}{2}\cos(x) & 1 \end{pmatrix} \tag{5}$$

Esempio Newton per sistemi

Successione Newton:

$$\begin{cases} f'(\mathbf{v}_k)) \cdot h_{k+1} = -f(\mathbf{v}_k), \\ \mathbf{v_{k+1}} := \mathbf{v_k} + h_{k+1} \end{cases} \tag{6}$$

x_k	y_k
0.000000000000000	0.000000000000000
0.500000000000000	0.250000000000000
0.486463513355199	0.233773076987732
0.486405155145713	0.233725502568820
0.486405154665921	0.233725501958721

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- **inline**: definisce funzione da valutare;

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- inline: definisce funzione da valutare;
- bisezione2011: funzione bisezione (da fare);

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- inline: definisce funzione da valutare;
- bisezione2011: funzione bisezione (da fare);
- a,b: intervallo iniziale bisezione (input);

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- inline: definisce funzione da valutare;
- bisezione2011: funzione bisezione (da fare);
- a,b: intervallo iniziale bisezione (input);
- tolres,tolintv,maxit: tolleranze bisezione (input);

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- inline: definisce funzione da valutare;
- bisezione2011: funzione bisezione (da fare);
- a,b: intervallo iniziale bisezione (input);
- tolres,tolintv,maxit: tolleranze bisezione (input);
- f: passaggio di funzione come variabile (input);

Bisezione in Matlab/Octave: `demobisezione2011.m`

Salviamo in `demobisezione2011.m` il seguente file

```
% f=inline('(x-1).^3'); a=0; b=1.5;
f=inline('x.^2-2'); a=1; b=2;
tolres=10^(-15);tolintv=10^(-15);maxit=10000;
[aa,bb]=bisezione(a,b,tolintv,tolres,maxit,f);
format long e; [aa bb]
```

- inline: definisce funzione da valutare;
- bisezione2011: funzione bisezione (da fare);
- a,b: intervallo iniziale bisezione (input);
- tolres,tolintv,maxit: tolleranze bisezione (input);
- f: passaggio di funzione come variabile (input);
- aa,bb: vettori intervalli analizzati da bisez. (output);

Bisezione in Matlab/Octave: criterio arresto

Il criterio del residuo $|f(c)| < \texttt{tol}$ non è spesso accettabile:

- funzione piatta: si pensi a risolvere l'equazione $f(x) = 0$ con $f(x) = 10^{-50} \cdot x$; il residuo $|f(1)| = 10^{-50}$ ma 1 è molto distante da $x^* = 0$.

Bisezione in Matlab/Octave: criterio arresto

Il criterio del residuo $|f(c)| < \texttt{tol}$ non è spesso accettabile:

- funzione piatta: si pensi a risolvere l'equazione $f(x) = 0$ con $f(x) = 10^{-50} \cdot x$; il residuo $|f(1)| = 10^{-50}$ ma 1 è molto distante da $x^* = 0$.
- funzione ripida: si pensi a risolvere l'equazione $f(x) = 10^{50} \cdot x = 0$; il residuo $|f(10^{-20})| = 10^{30}$ seppure 10^{-20} sia molto vicino a $x^* = 0$.

Bisezione in Matlab/Octave: criterio arresto

Siano $a < b$ e $c = (a+b)/2$. Diciamo residuo pesato $|f(c) \cdot w|$ con

$$w := \left(\frac{f(b) - f(a)}{b - a}\right)^{-1}.$$

Si vede subito che w^{-1} è un rapporto incrementale.

- funzione piatta: si pensi a risolvere l'equazione $f(x) = 0$ con $f(x) = 10^{-50} \cdot x$; w è grande ($w = 10^{50}$) e quindi $|f(1)| = 10^{-50}$ ma $|f(1)w| = 1$;
- funzione ripida: si pensi a risolvere l'equazione $f(x) = 0$ con $f(x) = 10^{50} \cdot x$; w è piccolo ($w = 10^{-50}$) e quindi $|f(10^{-20})| = 10^{30}$ ma $|f(10^{-20})w| = 10^{30} \cdot 10^{-50} = 10^{-20}$

Per f più generali, il test del residuo pesato $|f(c)w| <$ `tol` prova ad adattare situazioni in cui il test del residuo $|f(c)| <$ `tol` non sia affidabile.

Bisezione in Matlab/Octave: `bisezione.m`

```
function [aa,bb,ko]=bisezione(a,b,tinv,tres,maxit,f)
if b < a s=b; b=a; a=s; end % Aggiusta errori utente.
ko=0;  fa=feval(f,a); fb=feval(f,b); aa=[a]; bb=[b];
if fa*fb > 0 ko=1; return; end
if fa == 0 aa=[aa;a];bb=[bb;a];return;end % a sol.
if fb == 0 aa=[aa;b];bb=[bb;b];return;end % b sol.
for index=1:maxit
    c=(a+b)/2;  fc=feval(f,c);
    w=(b-a)/(fb-fa);wres=abs(fc*w); % Residuo pesato.
    semilung=(b-a)/2; % Semilung. intervallo.
    if (wres<tres)|(semilung<tinv)|(fc==0)
        aa=[aa;c];bb=[bb;c];return;% OK exit.
    end
    if sign(fc) == sign(fa) % Aggiornamento.
        aa=[aa; c];bb=[bb; b];a=c;fa=fc;
    else
        aa=[aa; a];bb=[bb; c];b=c;fb=fc;
    end
end
k0=1;
```

Bisezione in Matlab/Octave: esempio `demobisezione`

Usiamo `demobisezione` per il calcolo della soluzione positiva di $x^2 - 2 = 0$, cioè $\sqrt{2} = 1.41421356237309\ldots$

```
>> demobisezione2011
ans =
     1.000000000000000e+00     2.000000000000000e+00
     1.000000000000000e+00     1.500000000000000e+00
     1.250000000000000e+00     1.500000000000000e+00
     1.375000000000000e+00     1.500000000000000e+00
     ...
     1.414213562373092e+00     1.414213562373099e+00
     1.414213562373096e+00     1.414213562373096e+00
>> length(aa) % ITERAZIONI COMPIUTE.
ans = 49
>> sqrt(2)
ans = 1.414213562373095e+00
>>
```

Newton in Matlab/Octave: `demonewton2011.m`

Il file `demonewton2011.m` usa Newton per calcolo $\sqrt{2}$.

```
% f=inline('(x-1).^3'); x0=1.5;
f=inline('x.^2-2'); df=inline('2*x'); x0=1;
tolstep=10^(-15); maxit=1000;
[x,ko]=newton2011(x0,tolstep,maxit,f,df);
format long e; x
```

- Il comando inline permette di definire le funzioni $f(x) = x^2 - 2$ e $f^{(1)}(x) = 2x$, in forma vettoriale (si noti il .).

Newton in Matlab/Octave: `demonewton2011.m`

Il file `demonewton2011.m` usa Newton per calcolo $\sqrt{2}$.

```
% f=inline('(x-1).^3'); x0=1.5;
f=inline('x.^2-2'); df=inline('2*x'); x0=1;
tolstep=10^(-15); maxit=1000;
[x,ko]=newton2011(x0,tolstep,maxit,f,df);
format long e; x
```

- Il comando inline permette di definire le funzioni $f(x) = x^2 - 2$ e $f^{(1)}(x) = 2x$, in forma vettoriale (si noti il .).
- Il valore da cui parte il metodo di Newton è $x0$ (vicino a x^*).

Newton in Matlab/Octave: `demonewton2011.m`

Il file `demonewton2011.m` usa Newton per calcolo $\sqrt{2}$.

```
% f=inline('(x-1).^3'); x0=1.5;
f=inline('x.^2-2'); df=inline('2*x'); x0=1;
tolstep=10^(-15); maxit=1000;
[x,ko]=newton2011(x0,tolstep,maxit,f,df);
format long e; x
```

- Il comando inline permette di definire le funzioni $f(x) = x^2 - 2$ e $f^{(1)}(x) = 2x$, in forma vettoriale (si noti il .).
- Il valore da cui parte il metodo di Newton è $x0$ (vicino a x^*).
- tolstep definisce la tolleranza del criterio di arresto dello step

  $$|x_{k+1} - x_k| \leq \texttt{tolstep}$$

  uscendo al massimo dopo maxit iterazioni.

Newton in Matlab/Octave: `demonewton2011.m`

Il file `demonewton2011.m` usa Newton per calcolo $\sqrt{2}$.

```
% f=inline('(x-1).^3'); x0=1.5;
f=inline('x.^2-2'); df=inline('2*x'); x0=1;
tolstep=10^(-15); maxit=1000;
[x,ko]=newton2011(x0,tolstep,maxit,f,df);
format long e; x
```

- Il comando inline permette di definire le funzioni $f(x) = x^2 - 2$ e $f^{(1)}(x) = 2x$, in forma vettoriale (si noti il .).
- Il valore da cui parte il metodo di Newton è $x0$ (vicino a x^*).
- tolstep definisce la tolleranza del criterio di arresto dello step
$$|x_{k+1} - x_k| \leq \texttt{tolstep}$$
uscendo al massimo dopo maxit iterazioni.
- newton2011 è la routine che esegue il metodo di Newton. In output vettore iterazioni `x`, flag `ko` (0: ok, 1: ko).

Newton in Matlab/Octave: `newton2011.m`

```
function [x,ko]=newton2011(x0,tol,kmax,f,df)
x_old=x0; ko=0; x=[x0];
step=realmax; k=0;
while (abs(step) > tol) & (k < kmax)
    k=k+1; fx=feval(f,x_old);
    if fx == 0 return; end
    dfx=feval(df,x_old);
    if dfx == 0 ko=1; return; end
    step=-fx/dfx; x_new=x_old+step;
    x=[x; x_new]; x_old=x_new;
end
ko=1;
```

Newton in Matlab/Octave: esempio radice quadrata.

```
>> % NEWTON. CALCOLO RADICE QUADRATA.
>> demonewton2011
x =
     1.000000000000000e+00
     1.500000000000000e+00
     1.416666666666667e+00
     1.414215686274510e+00
     1.414213562374690e+00
     1.414213562373095e+00
     1.414213562373095e+00
>> diff(x)
ans =
     5.000000000000000e-01
    -8.333333333333326e-02
    -2.450980392156854e-03
    -2.123899820016817e-06
    -1.594724352571575e-12
    -2.220446049250313e-16
>>
```

Applicando il metodo di bisezione nell'intervallo $[1, 2]$ sarebbero servite 49 iterazioni.

Newton in Matlab/Octave: esempio $(x-1)^3$.

Risolviamo con il metodo di Newton l'equazione $(x-1)^3 = 0$ la cui soluzione è 1 con molteplicità 3 (zero multiplo!). Ci si aspetta convergenza solo lineare. Quale punto iniziale poniamo $x_0 = 1.5$. Il metodo offre la soluzione dopo 83 iterazioni.

```
>> demonewton2011
x =
     1.500000000000000e+00
     1.333333333333333e+00
     1.222222222222222e+00
     1.148148148148148e+00
     1.098765432098765e+00
     ...
     1.000000000000003e+00
     1.000000000000002e+00
>> length(x)
ans =
    83
>>
```

Bisezione in Matlab/Octave: esempio $(x-1)^3$.

Sullo stesso esempio studiamo bisezione, con $a = 0$ e $b = 1.5$. Il metodo offre la soluzione dopo 46 iterazioni.

```
>> demobisezione2011
ans =
      0.000000000000000e+00      1.500000000000000e+00
      7.500000000000000e-01      1.500000000000000e+00
      7.500000000000000e-01      1.125000000000000e+00
      9.375000000000000e-01      1.125000000000000e+00
      9.375000000000000e-01      1.031250000000000e+00
      9.843750000000000e-01      1.031250000000000e+00
      9.843750000000000e-01      1.007812500000000e+00
      ...
      9.999999999999432e-01      1.000000000000028e+00
      9.999999999999858e-01      1.000000000000028e+00
>> length(a)
ans =
    46
>>
```

Riferimenti bibliografici

- K. Atkinson, W. Han, *Elementary Numerical Analysis*, Wiley, (2004).
- A. Quarteroni, F. Saleri, *Introduzione al Calcolo Scientifico*, Springer, (2002).