Calcolo Scientifico - a.a. 2003/04
EQUAZIONI NON LINEARI

prof.ssa Rossana Vermiglio, dott. Dimitri Breda
Dipartimento di Matematica e Informatica
Università degli Studi di Udine
vermiglio,dbreda@dimi.uniud.it
http://www.dimi.uniud.it/∼rossana,dbreda

Indice

1 Appunti sulla teoria dei metodi per la risoluzione di equazioni non lineari

Il problema matematico che vogliamo risolvere numericamente è il seguente:

data una funzione $f : \mathcal{R} \rightarrow \mathcal{R}$ (scalare) trovare α tale che $f(\alpha) = 0$

α si chiama **radice** dell'equazione $f(x) = 0$ oppure **zero** della funzione f.

Il problema della ricerca degli zeri di una funzione ha sempre soluzione? La soluzione, se esiste, è unica?
Il problema può non avere soluzione (es. $f(x) = e^x + 2$); può avere unica soluzione (es. $f(x) = e^{-x} - x$); due soluzioni (es. $f(x) = x^2 - 1$); tre soluzioni (es. $f(x) = x^3 - 6x^2 + 11x - 6$);...; infinite soluzioni (es. $f(x) = cos(x)$).
Inoltre le radici possono essere semplici (i.e. $f'(\alpha) \neq 0$) o con molteplicità $\mu > 1$ (i.e. $f^{(k)}(\alpha) = 0, k = 1, \ldots, \mu - 1, f^{(\mu)}(\alpha) \neq 0$).

Come primo passo abbiamo bisogno di **localizzare la radice** α a cui siamo interessati, cercando un intervallo $[a, b]$ di ampiezza sufficientemente piccola che la contenga. Come fare? Lo strumento matematico che ci permette di risovere questa questione è dato dal seguente teorema:

Teorema 1 *Sia f una funzione continua in $[a, b]$ e tale che $f(a)f(b) < 0$ (segno opposto), allora f ha almeno uno zero α in (a, b).*

Un intervallo $[a, b]$ in cui la funzione f assume segno opposto agli estremi si chiama **intervallo di localizzazione** (bracket).
Come si può ridurre l'ampiezza di tale intervallo per "isolare" la radice con un'accuratezza prefissata?

1.1 Metodo di bisezione

L'idea del **metodo di bisezione** consiste nel dividere a metà l'intervallo di localizzazione $[a, b]$ ed invidiare quale delle due parti contiene la radice cercata, confrontando il segno della funzione nel punto medio con il segno della funzione in uno dei due estremi e utilizzando il Teorema 1. Si può ripetere più volte tale procedimento, dimezzando ad ogni passo l'ampiezza dell'intervallo di localizzazione, finchè non viene raggiunta l'accuratezza desiderata *tol*.
Assegnato in input un intervallo di localizzazione iniziale $[a, b]$, i valori $fa = f(a), fb = f(b)$ ed una prefissata precisione tol, l'algoritmo di bisezione puo' essere descritto come segue

```
if sign(fa)==sign(fb) then
    'attenzione segno concorde agli estremi'
    return
else
```

```
    while (|b-a| > tol) do
        m= a+(b-a)/2
        fm=f(m)
        if fm==0 then return
        if sign(fa)==sign(fm) then
            a=m
            fa=fm
        else
            b=m
            fb=fm
        end
    end
end
```

Tale algoritmo converge sempre, se applicato a funzioni continue. Il numero di iterazioni necessarie per ottenere un'intervallo di localizzazzione di ampiezza prefissata *tol* e' dato da

$$k = \text{intero-sup}\left(\log_2\left(\frac{|b-a|}{tol}\right)\right)$$

e non dipende dalla funzione considerata, ma solo dall'ampiezza dell'intervallo di localizzazzione iniziale. Per $|b-a| = 1$ e $tol = 10^{-7}$ sono necessarie $k =$intero-sup$(7 \times 3.32192) = 24$ iterazioni. In particolare per $tol = 10^{-1}$ servono $k = 4$ iterazioni: è un algoritmo piuttosto lento!

Vogliamo infine misurare la bontà dell' approssimazione di α ottenuta scegliendo $\tilde{\alpha} = m = (a+b)/2$. Il test dell' algoritmmo sopra descritto controlla l'errore assoluto, del quale mi fornisce la seguente stima

$$|\alpha - \tilde{\alpha}| \leq \frac{|b-a|}{2} \leq \frac{tol}{2}. \tag{1}$$

In analisi numerica, dove le quantità in gioco possono avere ordini di grandezza molto variabili, risulta in generale più opportuno controllare l'errore relativo. Modificando nell'algoritmo il test di controllo nel modo seguente

```
while ({|b-a|}/{min(|a|,|b|)} > tol) do
```

si ottiene la seguente stima dell'errore relativo

$$\frac{|\alpha - \tilde{\alpha}|}{|\alpha|} \leq \frac{|b-a|}{2\min\{|a|,|b|\}} \leq \frac{tol}{2}. \tag{2}$$

E' importante osservare che questo criterio può non funzionare se $\min\{|a|,|b|\}$ si avvicina a zero.

1.2 Metodo di iterazione funzionale

Un modo per costruire dei metodi alternativi per l'approssimazione degli zeri di funzione, che sarà utile in seguito anche per analizzare le proprietà di convergenza di particolari schemi di iterazione, nasce dalla riscrittura del problema

$$f(x) = 0 \tag{3}$$

come problema di **punto fisso**

$$x = g(x). \tag{4}$$

I problemi (3) e (4) sono **matematicamente equivalenti** se (3) e (4) hanno le stesse soluzioni in un intervallo che contiene la soluzione da approssimare. Assegnato un valore iniziale x_0 rimane definito lo schema iterativo (**metodo di iterazione funzionale**)

$$x_{k+1} = g(x_k), k = 0, 1, \ldots . \tag{5}$$

Se g è continua e la successione $\{x_k\}_{k\geq 0}$ definita da (5) è convergente, essa converge ad un punto fisso di g (i.e. $\alpha = g(\alpha)$) e quindi ad uno zero di f.

In generale, assegnato un problema (3), si possono scrivere più problemi di punto fisso (4) ad esso equivalenti. Le proprietà dei metodi di iterazione funzionale, descritti da diverse funzioni g, possono essere ben diverse, come si evince dal seguente esempio.

Esempio 1 *Data $f(x) = x^2 - x - 2$ si vuole approssimare la radice $\alpha = 2$.*

1. *Sia $g(x) = x^2 - 2$. Scelto $x_0 = 2.01$, si ottiene da (5) una successione non convergente al punto fisso (punto repulsivo), come dimostrano i risultati della Tabella 1.*

k	x_k	$\|e_k\|$
0	2.0100	5.0000e-03
1	2.0401e+00	2.0050e-02
2	2.1620e+00	8.1004e-02
3	2.6743e+00	3.3714e-01
4	5.1518e+00	5.1518e+00
5	2.4541e+01	11.1270e+01
6	6.0025e+02	2.9912e+02
7	3.6029e+05	1.8015e+05
8	1.2981e+11	6.4905e+10

Tabella 1: Approssimazione di $\alpha = 2$ con metodo 1

2. *Sia $g(x) = \sqrt{x+2}$. Scelto $x_0 = 1$, (5) fornisce una successione convergente in maniera monotana, come dimostrano i risultati in Tabella 2. E' importante notare che il rapporto fra gli errori in passi successivi $\frac{|e_{k+1}|}{|e_k|}$ tende a $1/4 = 2.5e - 01$. Ciò ci fornisce una misura della riduzione asintotica dell'errore e quindi della velocità del metodo stesso.*

k	x_k	$\|e_k\|$	$\|e_{k+1}\|/\|e_k\|$
0	1.0000000	5.0000e-01	
1	1.73205080	1.3397e-01	2.6795e-01
2	1.93185165	3.4074e-02	2.5433e-01
3	1.98288972	8.5551e-03	2.5107e-01
4	1.99571784	2.1411e-03	2.5027e-01
5	1.99892917	5.3541e-04	2.5007e-01
6	1.99973227	1.3386e-04	2.5002e-01
7	1.99993306	3.3466e-05	2.5000e-01
8	1.99998326	8.3666e-06	2.500041e-01
9	1.99999581	2.0916e-06	2.500002e-01
10	1.99999895	5.2291e-07	2.5000006e-01

Tabella 2: Approssimazione di $\alpha = 2$ con metodo 2

3. *Sia* $g(x) = 1 + \frac{2}{x}$. *Scelto* $x_0 = 1$, *si ottiene una successione convergente in maniera alternata, come dimostrano i risultati in Tabella 3. È importante notare come il rapporto fra gli errori in questo caso tenda ad* $1/2 = 5.0e-01$. *Il metodo risulta più lento del precedente.*

k	x_k	$\|e_k\|$	$\|e_{k+1}\|/\|e_k\|$
0	1.0000	5.0000e-01	
1	3.0000	5.0000e-01	1.0
2	1.6667	1.6667e-01	3.3333e-01
3	2.2000	1.0000e-01	6.0000e-01
4	1.9091	4.5455e-02	4.5455e-01
5	2.0476	2.3810e-02	4.8837e-01
6	1.9767	1.1628e-02	4.8837e-01
7	2.0118	5.8824e-03	5.0588e-01
8	1.9942	2.9240e-03	4.9708e-01
9	2.0029	1.4663e-03	5.0147e-01
10	1.9985	7.3206e-04	4.9927e-01

Tabella 3: Approssimazione di $\alpha = 2$ con metodo 3

4. *Sia* $g(x) = \frac{x^2+2}{2x-1}$. *Scelto* $x_0 = 1$, *si ottiene una successione convergente in maniera monotona, come dimostrano i risultati in Tabella 4. È importante notare come in questo caso l'errore tenda a zero molto rapidamente e che la dipendenza tra gli errori ai passi successivi risulta essere di tipo quadratico.*

Per cercare di analizzare il comportamento della successione definita da (5) abbiamo bisogno del seguente "strumento matematico":

Teorema 2 *Sia* g *una funzione differenziabile con continuità. Se* $g(\alpha) = \alpha$ *e* $|g'(\alpha)| < 1$ *allora esiste un intervallo* $I_\alpha = [\alpha - r, \alpha + r]$ *in cui, preso* $x_0 \in I_\alpha$, *l'iterazione funzionale (5) converge ad* α *(unico punto fisso).*

k	x_k	$\lvert e_k\rvert$	$\lvert e_{k+1}\rvert/\lvert e_k\rvert$
0	1.0000000	5.0000e-01	
1	3.0000000	5.0000e-01	1.0
2	2.20000000	1.0000e-01	2.0000e-01
3	2.01176470	5.8824e-03	5.8824e-02
4	2.00004578	2.2889e-05	3.8911e-03
5	2.0000000006942	3.4925e-10	1.5259e-05
6	2.00000000	0	0

Tabella 4: Approssimazione di $\alpha = 2$ con metodo 4

Questo risultato si ottiene ricordando che, per il teorema del valor medio, vale:

$$e_{k+1} = x_{k+1} - \alpha = g(x_k) - g(\alpha) = g'(\xi_k)(x_k - \alpha) = g'(\xi_k)e_k,$$

dove ξ_k e' un punto tra x_k e α. Pertanto è il comportamento della derivata di g in un opportuno intorno di α che determina la convergenza o meno della successione.

Si può inoltre osservare che se $\mathbf{0 < g'(\alpha) < 1}$, avremo localmente una **convergenza monotona**, mentre se $\mathbf{-1 < g'(\alpha) < 0}$ la **convergenza** sarà di tipo **alternato**.

Se $\mathbf{g'(\alpha) = 0}$ allora il metodo d'iterazione funzionale (5) sarà **sempre localmente convergente**.

Inoltre se $\mathbf{|g'(\alpha)| > 1}$ il punto fisso è **repulsivo** per il metodo d'iterazione funzionale (5)

Se $|g'(\alpha)| = 1$, ma esiste un intorno di α in cui $|g'(x)| < 1, x \neq \alpha$, allora l'iterazione funzionale (5) converge ad α ma molto lentamente.

Come misurare la velocità di convergenza di una successione? Mediante il concetto di **ordine di convergenza**.

1.2.1 Ordine di convergenza

Supponiamo di avere una successione $\{x_k\}_{k\geq 0}$ convergente ad α. Sia

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|} = l, \tag{6}$$

allora se

- $l = 1$ si parla di convergenza **sublineare**
- $0 < l < 1$ si parla di convergenza **lineare**
- $l = 0$ si parla di convergenza **superlineare**

È facile dimostrare che, se la successione $\{x_k\}_{k\geq 0}$ è definita da (5), risulta $l = |g'(\alpha)|$.

Nel caso di convergenza superlineare diventa interessante cercare di caratterizzare la riduzione dell'errore come segue. Se esiste un numero reale $\mathbf{p > 1}$ ed una costante L positiva tale che

$$\lim_{k\to\infty} \frac{|x_{k+1}-\alpha|}{|x_k-\alpha|^p} = L > 0. \tag{7}$$

allora si dice che la successione ha ordine di convergenza $\mathbf{p > 1}$.

Se la successione è definita da (5) e g ha le derivate continue fino all'ordine $p \geq 2$, p intero, in un intorno del punto fisso α, allora risulta che la successione ha ordine di convergenza p se e soltanto se

$$g'(\alpha) = g''(\alpha) = \ldots = g^{(p-1)}(\alpha) = 0, \quad g^{(p)}(\alpha) \neq 0. \tag{8}$$

Rissumendo possiamo concludere che: se $|g'(\alpha)| < 1$ allora esiste un intorno $I_\alpha = [\alpha - r, \alpha + r]$ tale che, preso $x_0 \in I_\alpha$, l'iterazione funzionale (5) converge ad α. Se $|g'(\alpha)| \neq 0$, allora la convergenza è lineare. Se vale (8) allora il metodo converge con ordine p.

1.2.2 Criteri di arresto

Rimane ancora da esaminare quali **criteri di arresto** proporre per interrompere l'iterazione (5) dopo un numero finito di passi, determinati da una richiesta di accuratezza prefissata. Assegnata una precisione *tol*, si può cercare di controllare l'errore assoluto o relativo rispettivamente con le condizioni

$$|x_{k+1} - x_k| < tol \tag{9}$$

e

$$\frac{|x_{k+1} - x_k|}{\min\{|x_{k+1}|, |x_k|\}} < tol, \tag{10}$$

se $|x_{k+1}|, |x_k| \neq 0$. Utilizzando (9) e (10) si ottengono rispettivamente le seguenti limitazioni

$$|x_k - \alpha| = \frac{|x_{k+1} - x_k|}{|g'(\xi_k) - 1|} < \frac{tol}{|g'(\xi_k) - 1|}$$

e

$$\frac{|x_k - \alpha|}{\alpha} = \frac{|x_{k+1} - x_k|}{|\alpha||g'(\xi_k) - 1|} < \frac{tol \min\{|x_{k+1}|, |x_k|\}}{|\alpha||g'(\xi_k) - 1|}.$$

In entrambi i casi risulta che l'errore commesso può essere tanto più grande della precisione *tol* fissata quanto più il valore $g'(\xi_k)$ è prossimo a 1. Tali test pertanto funzionano meglio per le successioni alternate, mentre possono non essere soddisfacenti se $g'(\alpha)$ è vicino a 1. Usando infine il criterio

$$|f(x_k)| < tol, \tag{11}$$

si ottiene che

$$|x_k - \alpha| = \frac{|f(x_k)|}{|f'(\epsilon)|} < \frac{tol}{|f'(\epsilon)|}.$$

Pertanto il criterio (11) può non essere soddisfacente se $|f'(\epsilon)|$ assume un valore prossimo a zero.

1.3 Metodo di Newton e metodo a pendenza costante. Metodo delle secanti

Una classe di metodi per la risoluzione di (3), può essere costruita definendo una funzione di punto fisso come segue

$$g(x) = x - \frac{f(x)}{m(x)}. \tag{12}$$

Il significato geometrico è chiaro: il valore dell'iterata x_{k+1} dato da

$$x_{k+1} = x_k - \frac{f(x_k)}{m(x_k)} \tag{13}$$

è l'intersezione con l'asse delle x della retta che passa per $(x_k, f(x_k))$ con coefficiente angolare $m(x_k)$.

Scegliendo $m(x) = f'(x)$ si ottiene **il metodo di Newton o delle tangenti**.

1.3.1 Metodo di Newton o delle tangenti

Supponiano che la funzione f sia sufficientemente regolare. Consideriamo dapprima il caso in cui α è una radice semplice (i.e. $f(\alpha) = 0, f'(\alpha) \neq 0$). Per studiare le proprietà di convergenza del metodo di Newton in base al Teorema 2, valutiamo il valore di g' in α. Poichè

$$g'(\alpha) = \frac{f(\alpha)f''(\alpha)}{f'(\alpha)^2} = 0, \tag{14}$$

il metodo risulta sempre localmente convergente in modo superlineare. Inoltre da

$$g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)}, \tag{15}$$

possiamo dedurre che il metodo convergerà quadraticamente se $f''(\alpha) \neq 0$, mentre convergerà ancora più velocemente se $f''(\alpha) = 0$.

Supponiamo ora che α sia una radice di molteplicità μ (i.e. $f(\alpha) = f'(\alpha) = \ldots = f^{(\mu-1)}(\alpha) = 0, f^{(\mu)}(\alpha) \neq 0$)). Allora possiamo riscrivere la funzione f come segue

$$f(x) = (x - \alpha)^\mu h(x),$$

dove h è una funzione continua tale che $h(\alpha) \neq 0$. Valutando g' in α, otteniamo ora che

$$g'(\alpha) = 1 - \frac{1}{\mu}. \tag{16}$$

Pertanto il metodo di Newton converge solo linearmente per le radici multiple! In particolare, per le radici doppie il fattore di riduzione asintotica dell'errore è $l = \frac{1}{2}$ (confrontabile con il metodo di bisezione).

Infine si può facilmente provare (esercizio) il seguente risultato:

Teorema 3 *Sia $J_\alpha = [\alpha, \alpha + r], r > 0$ (oppure $J_\alpha = [\alpha - r, \alpha], r > 0$) e $f \in C^2(J_\alpha)$. Se*

$$f(x)f''(x) > 0, f'(x) \neq 0, x \in J_\alpha, x \neq \alpha,$$

allora, preso $x_0 \in J_\alpha$, la successione definita dal metodo di Newton è convergente in maniera monotona.

Esercizio 1 *Per l'approssimazione della radice quadrata di un numero positivo a, i.e. $\alpha = \sqrt{a}$, si può risolvere il seguente problema $f(x) = x^2 - a = 0$. Studia la convergenza del metodo di Newton (vedi anche svolgimento Esercizio 5).*

Esercizio 2 *Per l'approssimazione del reciproco di un numero positivo a, i.e. $\alpha = \frac{1}{a}$, si può risolvere il seguente problema $f(x) = a - \frac{1}{x} = 0$. Scrivi la funzione di punto fisso per metodo di Newton e osserva che non sono coinvolte divisioni. Infine studia la convergenza.*

Il metodo di Newton richiede il calcolo della derivata $f'(x_k)$ ad ogni iterazione. In alcune situazioni questo non è consigliabile. In particolare quando il calcolo della derivata può essere molto costoso oppure la funzione f può essere il risultato di lunghe computazioni numeriche e la derivata non è pertanto descrivibile mediante una formula matematica. In questi casi si cerca di introdurre un'approssimazione $m(x_k)$ della derivata $f'(x_k)$ e i metodi (13) così ottenuti vengono chiamati **metodi quasi-Newton**. Ci sono molti metodi di questo tipo, il caso più semplice si ottiene ponendo $m(x) = m$ in (12) (**metodo a pendenza costante**).

1.3.2 Metodo a pendenza costante

Per il Teorema 2, per avere la convergenza del metodo a pendenza costante, dobbiamo imporre la condizione

$$|g'(\alpha)| = |1 - \frac{f'(\alpha)}{m}| < 1.$$

In particolare la convergenza è assicurata se valgono le seguenti condizioni

$$f'(x) \neq 0, x \in I_\alpha = [\alpha - r, \alpha + r]; mf'(x) > 0; |m| > \frac{1}{2} \max_{x \in I_\alpha} |f'(x)|.$$

Si potrebbe porre $m = f'(x_0)$, scegliendo $x_0 \in I_\alpha$ in modo che $|f'(x_0)| > \frac{1}{2} \max_{x \in I_\alpha} |f'(x)|$.

Se $f'(\alpha) \neq m$ il metodo converge linearmente, mentre ha convergenza superlineare se $f'(\alpha) = m$.

E' importante osservare che, a differenza del metodo di Newton che converge sempre localmente per le radici semplici, una scelta impropria del parametro di pendenza m può dare origine ad un metodo non convergente.

Esercizio 3 *Studia la convergenza del metodo a pendenza costante per il problema $f(x) = x^2 - a$.*

1.3.3 Metodo delle secanti

Il metodo della secanti appartiene alla classe dei metodi quasi-Newton ed approssima la derivata $f'(x_k)$ con il rapporto incrementale costruito con i valori delle due iterate precedenti.
L'iterazione è infatti descritta da

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, \ldots \tag{17}$$

dati x_0, x_1. L'interpretazione geometrica anche in questo caso è chiara: x_{k+1} è l' intersezione con l'asse delle x della retta secante costruita con i punti $(x_k, f(x_k)), (x_{k-1}, f(x_{k-1}))$. Una pseudocodofica del metodo edata da

```
while ({|x1-x0|} > tol*{min(|x1|,|x0|)}) do
t=x0
x0=x1
x1=x1+(x1-t)/(f(t)/f(x1)-1)
end
```

La convergenza del metodo delle secanti non può essere studiata mediante il Teorema 2, perchè la funzione d'iterazione dipende da due valori, i.e. $x_{k+1} = g(x_k, x_{k-1})$.

Si può dimostrare che l'errore soddisfa

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k||e_{k-1}|} = c \tag{18}$$

con $c > 0$. Questa relazione implica che il metodo è localmente convergente e suggerisce anche che la convergenza è superlineare ma non quadratica. Per determinare l'ordine di convergenza p si può procedere come segue. Sia $|e_{k+1}| \approx \gamma|e_k|^p$, allora da (18) si ottiene $|e_{k+1}| \approx c|e_k||e_{k-1}| \approx c\gamma^{\frac{-1}{p}}|e_k|^{1+\frac{1}{p}}$). Pertanto il metodo delle secanti ha ordine di convergenza

$$p = \frac{1+\sqrt{5}}{2} = 1.618\ldots,$$

che è la radice positiva dell'equazione $p = 1 + \frac{1}{p}$. Inoltre $\gamma = c^{\frac{1}{p}}$.

1.4 Metodo dell'interpolazione quadratica inversa

Il metodo delle secanti, utilizza la retta passante per due punti dell'iterata precedente per costruire l'approssimazione successiva. Si potrebbero usare tre valori x_k, x_{k-1}, x_{k-2} ed i corrispondenti $f(x_k), f(x_{k-1}), f(x_{k-2})$ per costruire la parabola che passa per tali punti e ottenere l'iterata x_{k+1} come intersezione della parabola con l'asse delle ascisse. Una difficoltà nasce dal fatto che possono non esserci radici reali. Questa osservazione può essere utilizzata per approssimare zei complessi della funzione $f(x)$ (**metodo di Muller**). In questo contesto non vogliamo usare l'artimetica complessa. Pertanto per ovviare al problema sopra esposto, il **metodo dell'interpolazione quadratica inversa** determina la parabola $p(y)$ nella variabile y che verifica le condizioni

$$p(f(x_k)) = x_k, p(f(x_{k-1})) = x_{k-1}, p(f(x_{k-2})) = x_{k-2}$$

e calcola x_{k+1} come intersezione conl'asse delle ascisse, i.e.

$$x_{k+1} = p(0).$$

L'algoritmo può essere descritto dalle seguenti formule che utilizzano la rappresentazione di Lagrange della parabola:

$$\begin{array}{l} u = \frac{f(x_{k-1})}{f(x_k)}, v = \frac{f(x_{k-1})}{f(x_{k-2})}, w = \frac{f(x_{k-2})}{f(x_k)}, \\ num = v(w(u-w)(x_k - x_{k-1}) - (1-u)(x_{k-1} - x_{k-2})) \\ den = (w-1)(u-1)(v-1) \\ x_{k+1} = x_k + \frac{num}{den}, \quad k = 2, 3, \ldots, \end{array} \tag{19}$$

dati x_0, x_1, x_2.

L'ordine di convergenza di tale metodo è $p \approx 1.839$ ed è lo stesso del metodo di Muller. Anche in questo caso la convergenza è locale e le iterazioni devono partire sufficientemente vicine alla soluzione α per garantire la convergenza dello schema.

Bisogna inoltre osservare che per applicare tale tecnica, i valori $f(x_k), f(x_{k-1}), f(x_{k-2})$ devono essere distinti e ciò non è sempre garantito.

1.5 Metodi affidabili

I metodi rapidamente convergenti come i metodi di Newton, secanti ed interpolazione quadratica inversa possono non convergere se non si hanno dei valori iniziali sufficientemente accurati. D'altro canto, un metodo sicuro come il metodo di bisezione è troppo lento. I metodi **ibridi** cercano di combinare le caratteristiche di entrambi i metodi. Si usa il metodo veloce in un intervallo di lacalizzazione della radice, ma se l'iterata esce dall'intervallo allora si ricorre alla bisezione per ridurre l'intervallo di localizzazione. Si riparte con il metodo veloce nell'intervallo più piccolo con più possibilità di successo.

1.6 Condizionamento del problema ed effetto errori di arrotandamento

Per studiare il condizionamento del problema consideriamo la funzione perturbata

$$\tilde{f}(x) = f(x) + e(x)$$

e supponiamo che l'errore $e(x)$ soddisfi la seguente limitazione

$$|e(x)| \leq \epsilon.$$

L'origine di tale errore può essere la più varia, potrebbe, per esempio, essere causata dagli errori di arrotondamento nel calcolo della funzione f. Indichiamo con β la radice del problema perturbato, i.e. $\tilde{f}(\beta) = 0$. In prima approssimazione si ottiene

$$f(\beta) = f(\alpha) + f'(\alpha)(\beta - \alpha) = -e(\beta),$$

da cui, assumendo $f'(\alpha) \neq 0$,

$$|\beta - \alpha| = \frac{|e(\beta)|}{|f'(\alpha)|} \leq \frac{\epsilon}{|f'(\alpha)|}.$$

Il parametro che misura il condizionamento del problema è $\frac{1}{|f'(\alpha)|}$ e pertanto il problema risulta ben condizionato per gli zeri α con derivata grande, mentre per valori della derivata prossimi a zero il problema risulta mal condizionato. Sia $[a, b]$ un intervallo che contiene α tale che

$$|f(x)| \leq \epsilon, \quad x \in [a, b].$$

Finchè siamo fuori da questo intervallo il valore $\tilde{f}(x)$ ci fornisce informazioni utili sulla localizzazione dello zero perchè le due funzioni $\tilde{f}, f$ hanno lo stesso segno. All'interno di tale intervallo invece il segno di $\tilde{f}$ può essere positivo, negativo o zero senza nessuna relazione con il segno di f. L'ampiezza di tale intervallo dipende dal condizionamento del problema: se è piccola, il problema è ben condizionato, se è grande il problema è mal condizionato. Un buon algoritmo (i.e. un algoritmo stabile) ci fornirà come risposta un punto in tale intervallo e non possiamo aspettarci di più. Un algoritmo stabile risolverà un problema ben condizionato in maniera accurata, ma nessun algoritmo potrà darci una soluzione accurata di un problema malcondizionato.

Analizziamo ora l'effetto degli errori di arrotondamento nel calcolo delle iterate del metodo (5). Sia $\tilde{x}_{k+1}$ il valore effettivamente calcolato e supponiamo che valga inoltre

$$\tilde{x}_{k+1} = g(\tilde{x}_k) + \delta_k,$$

con

$$|\delta_k| \leq \delta, k = 0, 1, 2, \ldots.$$

Nelle ipotesi del teorema 2 si può dimostrare che, preso $\tilde{x}_0 \in I_\alpha$, vale

$$|\tilde{x}_k - \alpha| \leq \lambda^k |\tilde{x}_0 - \alpha| + \sigma(1 - \lambda^k),$$

dove $\lambda := \max\limits_{x \in I\alpha} |g'(x)|$ e $\sigma := \frac{\delta}{1-\lambda}$. Pertanto σ ci fornisce una misura dell'incertezza con cui è possibile determinare la soluzione per effetto degli errori di arrotantamento. Analogo risultato vale per l'errore relativo.

2 Istruzioni MATLAB

2.1 fzero

La function *fzero* cerca di approssimare lo zero di una funzione scalare di variabile reale più vicino al valore iniziale fornito dall'utente. La sintassi è la seguente:

```
[x,fval]=fzero(fun,x0,options,p1,p2,...)
```

dove x è lo zero trovato, $fval$ il residuo, corrispondente all'istruzione

```
fval=feval(fun,z)
```

fun è una function passata tra apici (quindi una function predefinita o un oggetto *inline*), $x0$ è un valore scalare (o un intervallo se è un vettore di lunghezza due, questo evita di cadere nelle singolarità) iniziale, *options* è un insieme di opzioni definite dall'utente e $p1$, $p2$, etc. un'eventuale serie di parametri contemplati da fun.

options definisce la modalità di visualizzazione e la tolleranza richiesta. Viene impostato nel seguente modo:

```
options=optimset('Display','D','TolX','T')
```

dove D=

- *off*: non viene visualizzato niente;
- *iter*: viene visualizzato il risultato di ogni iterazione;
- *final*: viene visualizzato il risultato dell'ultima iterazione;
- default=*final*

e T è la tolleranza fissata dall'utente (default=*eps*).

L'algoritmo di ricerca si basa su una combinazione dei metodi di bisezione, delle secanti e di interpolazione quadratica inversa. In particolare, l'algoritmo parte dal punto iniziale (o dall'intervallo) $x0$ fornito dall'utente e cerca, alternativamente a sinistra e a destra con spostamenti sempre più ampi, un punto in cui la funzione fun cambia segno rispetto al punto iniziale. Si determina così un intervallo al cui interno c'è uno zero di fun. A questo punto l'algoritmo procede con un passo del metodo delle secanti, che richiede due valori iniziali, per determinarne un terzo e poter così procedere con l'interpolazione quadratica inversa, che invece ne richiede tre. Quest'ultimo metodo converge più velocemente delle secanti. Infine, se ad un passo l'algoritmo incontra un cambio di segno di fun rispetto all'iterata precedente, allora viene applicato un passo di bisezione che dimezza l'intervallo di ricerca e poi riprende come sopra descritto.

2.2 fsolve

La function *fsolve* è l'analoga di *fzero* per risolvere sistemi di equazioni non lineari, i.e. funzioni vettoriali di più variabili. E' disponibole con il toolbox di ottimizzazione. La sintassi è la medesima vista per *fzero*, ma gli algoritmi di ricerca su cui si basa sono completamente diversi (ad esempio metodi quasi-Newton, gradiente coniugato, line-search, etc.).

2.3 Polinomi

Un polinomio di grado n

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$

viene identificato in MATLAB con il vettore di $n+1$ elementi

```
p=[p_1,p_2,...,p_(n+1)]
```

dei suoi coefficienti ordinati da quello corrispondente al termine di grado maggiore fino al termine noto.

La valutazione del polinomio p in un punto x avviene con l'istruzione

```
y=polyval(p,x)
```

la quale utilizza l'algoritmo di Horner.

La function *roots* serve per il calcolo del vettore z delle n radici del polinomio:

```
z=roots(p)
```

e si basa sul calcolo degli autovalori della corrispondente matrice companion

$$\begin{pmatrix} -\frac{p_2}{p_1} & -\frac{p_3}{p_1} & \cdots & \cdots & -\frac{p_{n+1}}{p_1} \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

Dato infine un vettore z di lunghezza n, l'istruzione

```
p=poly(z)
```

ricostruisce il vettore p dei coefficienti del polinomio monico (i.e. $p_1 = 1$) di grado n che ha come radici gli elementi di z.

NOTA. Per tutti i comandi precedenti si consiglia di consultare l'*help* e la documentazione (*doc*) in linea di MATLAB.

3 Esercizi svolti

Esercizio 4 *Si vogliono approssimare le radici dell'equazione*

$$f(x) = x^2 - kx + 1 = 0, \quad k > 2,$$

utilizzando i tre metodi di iterazione funzionale

$$\begin{aligned} x_{i+1} &= g_1(x_i) = \frac{1}{k - x_i} \\ x_{i+1} &= g_2(x_i) = \frac{1 + x_i^2}{k} \\ x_{i+1} &= g_3(x_i) = k - \frac{1}{x_i}. \end{aligned}$$

Studiare la convergenza dei tre metodi e fornire delle scelte del valore iniziale che la garantiscono.

Soluzione. Utilizzando la formula risolutiva per le equazioni di secondo grado si ottengono le due radici reali

$$\alpha = \frac{k - \sqrt{k^2 - 4}}{2}, \quad \beta = \frac{k + \sqrt{k^2 - 4}}{2}$$

e, per $k > 2$, risulta $\alpha < 1 < k/2 < \beta$. Dopo aver valutato che i problemi di punto fisso g_1, g_2 e g_3 sono equivalenti al problema di trovare gli zeri di f, per studiare la convergenza dei tre metodi si calcola la derivata prima delle funzioni g_j, $j = 1, 2, 3$:

$$\begin{aligned} g_1'(x) &= \frac{1}{(k-x)^2}, \quad x \neq k \\ g_2'(x) &= \frac{2x}{k} \\ g_3'(x) &= \frac{1}{x^2}, \quad x \neq 0 \end{aligned}$$

e si valuta per $x = \alpha$ e $x = \beta$:

$$\begin{aligned} g_1'(\alpha) = \frac{1}{\beta^2} \in (0,1) \quad &, \qquad g_1'(\beta) = \frac{1}{\alpha^2} > 1 \\ g_2'(\alpha) = 1 - \sqrt{1 - \frac{4}{k^2}} \in (0,1) \quad &, \qquad g_2'(\beta) = 1 + \sqrt{1 - \frac{4}{k^2}} > 1 \\ g_3'(\alpha) = \frac{1}{\alpha^2} > 1 \quad &, \qquad g_3'(\beta) = \frac{1}{\beta^2} \in (0,1) \end{aligned}$$

Scegliendo opportunamente i valori iniziali, si conclude che g_1 e g_2 convergono alla radice minore α mentre g_3 converge a quella maggiore β, tutti in modo lineare monotono.

Una condizione sufficiente per determinare delle scelte del valore iniziale che garantiscono la convergenza è data dal Teorema 2, ovvero $x_0 \in I_\alpha$ con $|g'(x)| < 1$ $\forall x \in I_\alpha$ per g_1 e g_2 o $x_0 \in I_\beta$ con $|g'(x)| < 1$ $\forall x \in I_\beta$ per g_3. Nel primo caso risulta dunque

$$|g_1'(x)| = \frac{1}{(k-x)^2} < 1 \Rightarrow x < k - 1 \vee x > k + 1$$

e poichè $\alpha < 1$ si sceglie $x_0 < k - 1$. Nel secondo caso

$$|g_2'(x)| = \frac{2|x|}{k} < 1 \Rightarrow -\frac{k}{2} < x < \frac{k}{2}$$

per cui si sceglie $x_0 \in (-\frac{k}{2}, \frac{k}{2})$. Infine, nel terzo caso

$$|g_3'(x)| = \frac{1}{x^2} < 1 \Rightarrow x < -1 \vee x > 1$$

per cui si sceglie $x_0 > 1$ dato che $\beta > 1$. ■

Esercizio 5 *(vedi anche* 09/12/02$-\sharp$3*) Si vuole approssimare la radice positiva* $\alpha = \sqrt{a}$ *dell'equazione*

$$f(x) = x^2 - a = 0$$

utilizzando il metodo di Newton. Si valuti la convergenza dello schema iterativo ottenuto e si fornisca una scelta del valore iniziale che la garantisce. Cosa succede scegliendo x_0 *molto grande (i.e.* $x_0 = 10^6$*)?*

Soluzione. Applicando il metodo di Newton ad f, l'algoritmo iterativo risulta

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right) = g(x_k). \tag{20}$$

L'ordine di convergenza del metodo iterativo g è definito da (6) e (7). In particolare, essendo α radice semplice, da (14) segue $l = 0$, per cui il metodo è superlineare. Inoltre da (15) risulta $g''(\alpha) = 1/\alpha \neq 0$, per cui il metodo converge con ordine $p = 2$ e la costante in (7) vale $L = \frac{1}{2\alpha}$.

Usando il Teorema 3 si può dire che il metodo converge per ogni $x_0 > \alpha$. Si può inoltre verificare che per $x_0 \in (0, \alpha)$, $x_1 > \alpha$, per cui si conclude che il metodo converge per ogni $x_0 > 0$.

Implementando l'algoritmo con MATLAB si può ottenere una stima dell'ordine di convergenza con l'istruzione

```
p=log((1/L)*abs(e(2:length(e))))./log(abs(e(1:length(e)-1)));
```

che calcola

$$p = \frac{log\,(|e_{k+1}|/L)}{log|e_k|}$$

una volta noto il fattore di convergenza L.

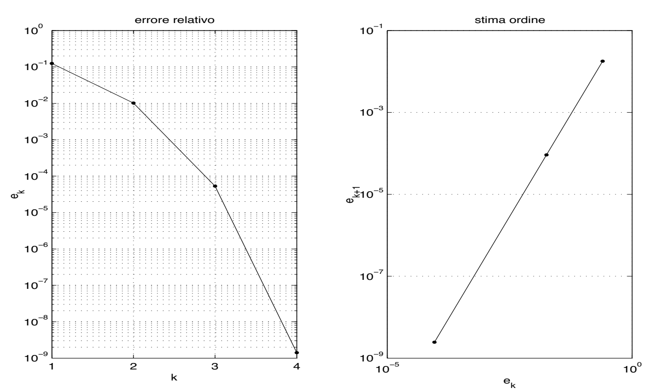

Figura 1: Errore relativo e stima dell'ordine di convergenza per $x_0 = 2$ e $\alpha = 3$.

Nel caso $\alpha = 3$, partendo da un valore iniziale $x_0 = 2$ si ottengono i valori riportati in tabella 5 e in figura 1 che confermano l'analisi svolta.

k	x_k	$\|e_k\|$	p
1	1.750000000000000e+000	2.500000000000000e-001	2.007436835848513e+000
2	1.732142857142857e+000	1.785714285714286e-002	2.000013202183833e+000
3	1.732050810014728e+000	9.204712812970961e-005	1.999999996076083e+000
4	1.732050807568877e+000	2.445850337895745e-009	

Tabella 5: Approssimazione di $\alpha = \sqrt{3}$, errore assoluto e stima dell'ordine di convergenza per $x_0 = 2$ ($TOL = 10^{-5}$).

Vediamo invece cosa succede partendo da un valore iniziale $x_0 = 10^6$. Nel caso $\alpha = 1$ si ottengono i valori riportati in tabella 6 e in figura 2. Come si può osservare in figura 2 a destra, il metodo è quadratico (pendenza $p = 2$) nelle ultime iterazioni, mentre converge inizialmente in modo pressochè lineare (pendenza $p = 1$). Il fenomeno si spiega con l'analisi del metodo iterativo

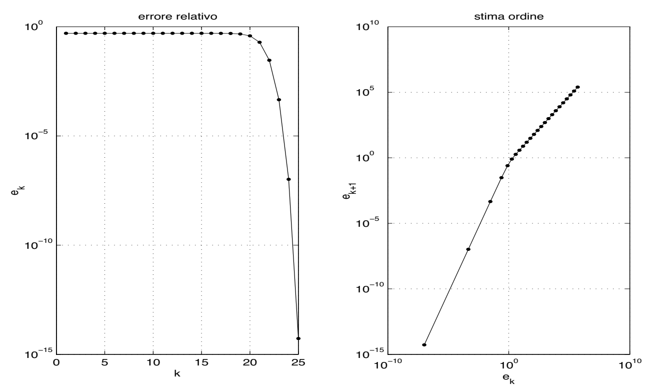

Figura 2: Errore relativo e stima dell'ordine di convergenza per $x_0 = 10^6$ e $\alpha = 1$ ($TOL = 10^{-10}$).

g definito da (20). Fintantochè x_k e $\frac{1}{x_k}$ non hanno un'ordine di grandezza confrontabile il metodo iterativo risulta un semplice dimezzamento. Infatti per $x_k \gg \frac{1}{x_k}$ si ha

$$g(x_k) \simeq \tilde{g}(x_k) = \frac{1}{2}x_k$$

e lo si vede chiaramente dai primi valori in tabella 1. In questa fase il metodo è quindi lineare poichè $\tilde{g}'(\alpha) = \frac{1}{2} \neq 0$. Quando x_k diventa dell'ordine dell'unità il fenomeno cessa e il metodo diventa quadratico come previsto dalla teoria. Per stimare il numero di iterazioni affinchè questo avvenga basta osservare che

$$x_k = \frac{1}{2}x_{k-1} = \frac{1}{2^2}x_{k-2} = \cdots = \frac{1}{2^k}x_0,$$

per cui posto $x_k = 1$, partendo con $x_0 = 10^6$ segue

$$k = log_2 \frac{x_0}{x_k} \simeq 20,$$

valore che conferma i risultati ottenuti.

k	x_k	$\|e_k\|$	p
1	5.000000000005000e+005	4.999999999995000e+005	9.999999999998476e-001
2	2.500000000012500e+005	2.499999999992500e+005	9.999999999993563e-001
3	1.250000000026250e+005	1.249999999986250e+005	9.999999999972734e-001
4	6.250000000531250e+004	6.249999999731250e+004	9.999999999884088e-001
5	3.125000001065625e+004	3.124999999465625e+004	9.999999999505304e-001
6	1.562500002132813e+004	1.562499998932813e+004	9.999999997879175e-001
7	7.812500042664063e+003	7.812499978664063e+003	9.999999990860693e-001
8	3.906250085332031e+003	3.906249957332032e+003	9.999999960378864e-001
9	1.953125170666013e+003	1.953124914666018e+003	9.999999827017582e-001
10	9.765628413329839e+002	9.765623293330286e+002	9.999999238400683e-001
11	4.882819326663130e+002	4.882809086666709e+002	9.999996612522208e-001
12	2.441419903317248e+002	2.441399423345881e+002	9.999984741762777e-001
13	1.220730431544093e+002	1.220689471773156e+002	9.999930162439539e-001
14	6.104061748558112e+001	6.103242566882813e+001	9.999673562409636e-001
15	3.052850000986151e+001	3.051211747571961e+001	9.998429621398842e-001
16	1.528062814338625e+001	1.524787186647526e+001	9.992123341963466e-001
17	7.673035239652204e+000	7.607592903734042e+000	9.957787918714165e-001
18	3.901680881088885e+000	3.771354358563319e+000	9.744068123196803e-001
19	2.078990336765699e+000	1.822690544323186e+000	7.808335481022778e-001
20	1.279996526738297e+000	7.989938100274016e-001	3.100067183009171e+000
21	1.030624323327378e+000	2.493722034109187e-001	2.021719879520034e+000
22	1.000454990804133e+000	3.016933252324524e-002	2.000129933306780e+000
23	1.000000103461242e+000	4.548873428911238e-004	2.000000013462164e+000
24	1.000000000000005e+000	1.034612366429416e-007	2.000268261662743e+000
25	1.000000000000000e+000	5.329070518200723e-015	

Tabella 6: Approssimazione di $\alpha = 1$, errore assoluto e stima dell'ordine di convergenza per $x_0 = 10^6$ $(TOL = 10^{-10})$.

NOTA. Si osservi che vale la seguente relazione

$$x_{k+1}^2 - a = \frac{(x_k^2 - a)^2}{4x_k^2}.$$

Infatti, utilizzando (20):

$$\begin{aligned} x_{k+1}^2 - a &= \frac{1}{4}\left(x_k + \frac{a}{x_k}\right)^2 - a = \\ &= \frac{1}{4}\left(x_k^2 + 2a + \frac{a^2}{x_k^2}\right) - a = \\ &= \frac{1}{4}\left(x_k^2 - 2a + \frac{a^2}{x_k^2}\right) = \\ &= \frac{(x_k^2 - a)^2}{4x_k^2}. \end{aligned}$$

La precedente puè essere utilizzata per la valutazione dell'ordine di convergenza del metodo. Essa può esere infatti riscritta come

$$\frac{e_{k+1}}{(e_k)^2} = \frac{x_{k+1}-\alpha}{(x_k-\alpha)^2} = \frac{(x_k+\alpha)^2}{x_{k+1}+\alpha} \cdot \frac{x_{k+1}^2-a}{(x_k^2-a)^2} = \frac{(x_k+\alpha)^2}{x_{k+1}+\alpha} \cdot \frac{1}{4x_k^2}$$

che tende a $\frac{1}{2\alpha} = L = \frac{1}{2}|g''(\alpha)|$, per $k \to \infty$ per cui si conferma che il metodo è quadratico. ■

Esercizio 6 *Si vuole approssimare l'unica radice* $\alpha = 1$ *di molteplicità* $\mu = 4$ *dell'equazione*

$$f(x) = (x-1)^4 = 0 \tag{21}$$

utilizzando il metodo di Newton. Si valuti la convergenza dello schema iterativo ottenuto e si fornisca una scelta del valore iniziale che la garantisce. Cosa succede se si implementa il metodo di Newton per il problema equivalente

$$f(x) = x^4 - 4x^3 + 6x^2 - 4x + 1 = 0? \tag{22}$$

Soluzione. L'algoritmo iterativo per (21) risulta

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k-1}{4} = \frac{3x_k+1}{4} = g(x_k).$$

L'ordine di convergenza p del metodo iterativo g è definito da (6) e (7). Poichè

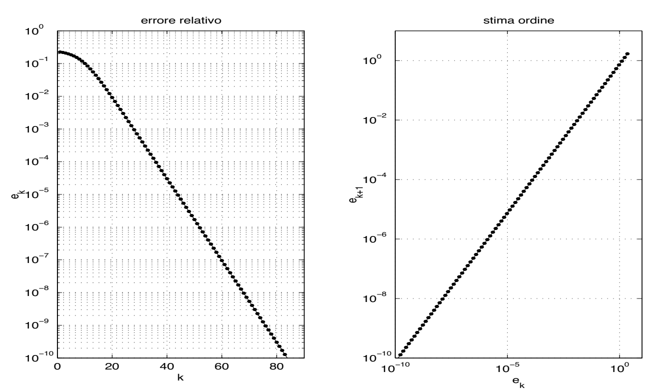

Figura 3: Errore relativo e stima dell'ordine di convergenza per $x_0 = 10$ e per la formulazione (21) ($TOL = 10^{-10}$).

g deriva dall'applicazione del metodo di Newton ad f e la radice $\alpha = 1$ ha molteplicità $\mu = 4$, segue da (16)

$$g'(\alpha) = 1 - \frac{1}{\mu} = \frac{3}{4}.$$

Secondo (6), il metodo converge dunque in modo lineare ($p = 1$) e la costante vale $l = \frac{3}{4}$. Applicando poi il Teorema 3 segue

$$f(x)f''(x) = (x-1)^4 \cdot 12(x-1)^2 > 0 \Rightarrow \forall x \neq 0$$

ed essendo $\alpha = 1 > 0$ il metodo converge scegliendo $x_0 > 0$.

Implementando l'algoritmo con MATLAB si può ottenere una stima dell'ordine di convergenza analogamente a quanto fatto nel precedente esercizio 5. Partendo da un valore iniziale $x_0 = 10$ si ottengono i valori di p plottati in figura 3 a destra che confermano la convergenza lineare.

Se invece si implementa lo stesso algoritmo per la formulazione estesa (22) si ottengono i valori riportati in figura 4. La convergenza è lineare, ma la tolleranza richiesta $TOL = 10^{-10}$ non è raggiunta in quanto con la formulazione precedente interviene il fenomeno della cancellazione nella valutazione di f. Quando infatti x_k raggiunge un valore accurato alla quarta cifra significativa e prossimo alla quinta (per $k = 47$ si ha $x_k = 1.000146585188209$), all'iterazione successiva la cancellazione è totale e il metodo si arresta non raggiungendo la tolleranza richiesta, ma fermandosi a circa 10^{-4}. ■

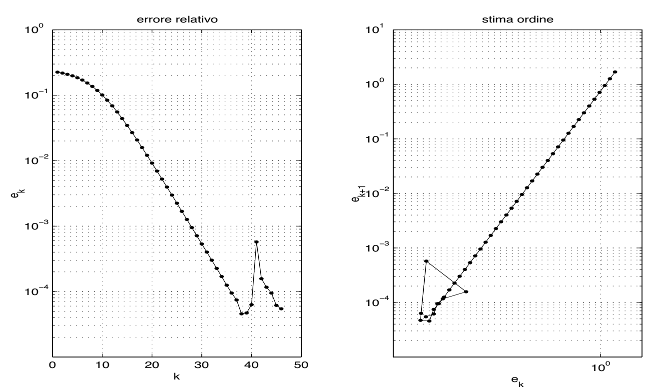

Figura 4: Errore relativo e stima dell'ordine di convergenza per $x_0 = 10$ e per la formulazione (22) ($TOL = 10^{-10}$).

Esercizio 7 *(vedi anche* 04/07/02 $-\sharp 4$*) Si vuole applicare il metodo di Newton per la ricerca dell'unica radice reale α dell'equazione*

$$f(x) = x^3 - x - 5 = 0.$$

Si localizzi tale radice, si valuti la convergenza dello schema iterativo ottenuto e si fornisca una scelta del valore iniziale che la garantisce. Il metodo di iterazione funzionale

$$h(x) = x^3 - 5$$

converge alla radice α?

Soluzione. La radice viene localizzata cercando due valori a e b tali che $f(a)f(b) < 0$. Ad esempio per $a = 1$ e $b = 2$ si ottiene $f(a) = -5 < 0$ e $f(b) = 1 > 0$, per cui $\alpha \in (1, 2)$. Lo schema iterativo risulta

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^3 - x_k - 5}{3x_k^2 - 1} = \frac{2x_k^3 + 5}{3x_k^2 - 1} = g(x_k), \quad x_k \neq \pm\frac{1}{\sqrt{3}}.$$

Essendo il metodo iterativo g derivato dall'applicazione del metodo di Newton e la radice $\alpha \in (1, 2)$ semplice (lo si deduce dal fatto che $f'(x) = 0 \Leftrightarrow x = \pm 1/\sqrt{3}$), da (14) e (15) l'ordine di convergenza è $p = 2$.

In figura 5 e tabella 7 sono riportati i valori relativi all'implementazione MATLAB del metodo di Newton g partendo con $x_0 = 2$. Si consideri ora il

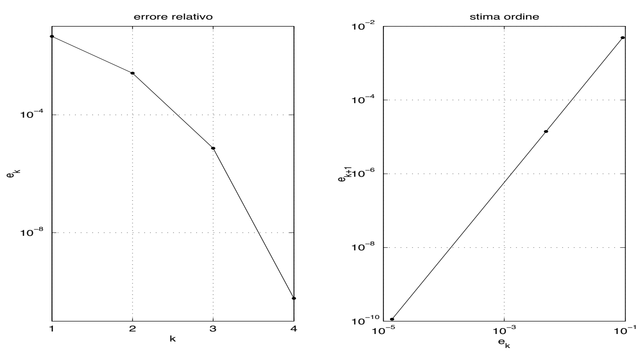

Figura 5: Errore relativo e stima dell'ordine di convergenza per il metodo di Newton g con $x_0 = 2$.

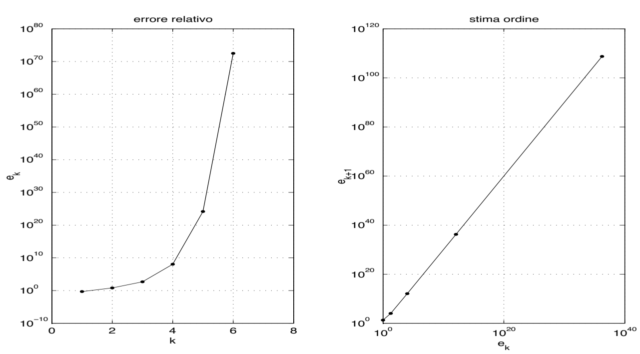

Figura 6: Errore relativo e stima dell'ordine di convergenza per il metodo di iterazione funzionale h con $x_0 = 2$.

metodo di iterazione funzionale h. Per tale metodo risulta

$$h'(\alpha) = 3x^2_{|x \in (1,2)} \in (3, 12)$$

k	x_k	$\lvert e_k \rvert$	p
		Metodo di Newton g	
1	1.909090909090909e+000	9.091e-002	1.988262740055438e+000
2	1.904174860081682e+000	4.916e-003	1.999678128682720e+000
3	1.904160859248288e+000	1.400e-005	1.999999437513644e+000
4	1.904160859134921e+000	1.134ie-010	
		Metodo di iterazione funzionale h	
1	3.000000000000000e+000	1.000000000000000e+000	Inf
2	2.200000000000000e+001	1.900000000000000e+001	3.148507582570453e+000
3	1.064300000000000e+004	1.062100000000000e+004	3.000669608877375e+000
4	1.205569317702000e+012	1.205569307059000e+012	3.000000000952068e+000
5	1.752171287496633e+036	1.752171287496633e+036	3.000000000000000e+000
6	5.379348465181141e+108	5.379348465181141e+108	Inf
7	Inf	Inf	

Tabella 7: Approssimazione di α, errore assoluto e stima dell'ordine di convergenza per $x_0 = 2$ ($TOL = 10^{-5}$).

per cui è sicuramente $h'(\alpha) > 1$, motivo per cui non si ha convergenza per questo schema. In figura 6 e tabella 7 sono riportati i valori relativi all'implementazione MATLAB del metodo di iterazione funzionale h partendo con $x_0 = 2$. ■

Esercizio 8 *(vedi anche* 10/07/03 – ♯5*) L'equazione non lineare*

$$f(x) = x^2 - x - 2 = 0$$

ha due radici $\alpha_1 = -1$ *e* $\alpha_2 = 2$. *Per approssimare la radice* α_2, *si propongono i seguenti metodi di iterazione funzionale*

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

con

1. $g_1(x) = x^2 - 2$
2. $g_2(x) = \sqrt{x+2}$
3. $g_3(x) = 1 + \frac{2}{x}$
4. $g_4(x) = \frac{x^2+2}{2x-1}$.

Si analizzi la convergenza dei metodi proposti.

Soluzione. Per analizzare la convergenza dei metodi proposti si calcola la derivata prima $g_j'(\alpha_2)$, $j = 1, 2, 3, 4$:

1. $g_1'(\alpha_2) = 2x_{|x=\alpha_2} = 4 > 1 \rightarrow$ il metodo non converge;
2. $g_2'(\alpha_2) = \frac{1}{2\sqrt{x+2}}_{|x=\alpha_2} = \frac{1}{4} \in (0, 1) \rightarrow$ il metodo converge in modo lineare monotono;

3. $g_3'(\alpha_2) = -\frac{2}{x^2}_{|x=\alpha_2} = -\frac{1}{2} \in (-1, 0) \rightarrow$ il metodo converge in modo lineare alternato;
4. $g_4'(\alpha_2) = \frac{2x(2x-1)-2(x^2+2)}{2x-1} = \frac{2x^2-2x-4}{2x-1}_{|x=\alpha_2} = 0 \rightarrow$ il metodo converge in modo superlineare.

In particolare, come si può facilmente verificare, il metodo g_4 corrisponde al metodo di Newton applicato ad f. È chiaro quindi che converge in modo almeno quadratico essendo la radice α_2 semplice. ■

Esercizio 9 *(vedi anche 24/07/03 – ♯4) Per calcolare le radici dell'equazione non lineare*

$$f(x) = 1 - \frac{1}{x} = 0$$

si consideri il seguente metodo iterativo (metodo di Steffensen):

$$x_{i+1} = x_i - \frac{f(x_i)^2}{f(x_i + f(x_i)) - f(x_i)}, \quad i = 0, 1, \ldots.$$

Si studi la convergenza e l'ordine di convergenza di tale metodo considerato come metodo di punto fisso.

Soluzione. Ricaviamo lo schema di iterazione funzionale:

$$\begin{aligned} g(x_i) &= x_{i+1} = x_i - \frac{f(x_i)^2}{f(x_i + f(x_i)) - f(x_i)} = \\ &= x_i - \frac{\left(1 - \frac{1}{x_i}\right)^2}{1 - \frac{1}{x_i + 1 - \frac{1}{x_i}} - \left(1 - \frac{1}{x_i}\right)} = \\ &\ldots \\ &= -\frac{x_i^3 - x_i^2 - 2x_i + 1}{x_i}. \end{aligned}$$

Si verifica facilmente che g è di punto fisso, risulta infatti $g(\alpha) = \alpha$ con $\alpha = 1$ radice da approssimare. Per valutare la convergenza si calcola la derivata prima:

$$g'(\alpha) = -\frac{x(3x^2 - 2x - 2) - x^3 + x^2 + 2x - 1}{x^2}_{|x=\alpha} = -\frac{2x^3 - x^2 - 1}{x^2}_{|x=\alpha} = 0$$

per cui il metodo converge in modo almeno quadratico. In particolare, calcolando la derivata seconda

$$g''(\alpha) = -\frac{x^2(6x^2 - 2x) - 2x(2x^3 - x^2 - 1)}{x^4}_{|x=\alpha} = -\frac{2x^3 + 2}{x^3}_{|x=\alpha} = -4 \neq 0$$

si può concludere che il metodo converge con ordine $p = 2$.

In generale, se α è radice semplice di $f(x) = 0$, allora il metodo di *Steffensen* converge con ordine $p = 2$ se $f'(\alpha) \neq -1$ e $f''(\alpha) \neq 0$, mentre converge con ordine $p = 3$ se $f'(\alpha) = -1$ o $f''(\alpha) = 0$. ■

Esercizio 10 *(vedi anche HW01 2002/03 – ♯5) Si scriva e si esegua una function MATLAB che associ al parametro m, $m \in [0, +\infty)$, la più piccola radice positiva dell'equazione*

$$\sin x = mx$$

utilizzando la function predefinita fzero. Si studi attentamente la scelta dei valori iniziali affinchè fzero converga alla radice positiva più piccola per $m = 0, 0.01, 0.02, \ldots, 0.98, 0.99, 1, 2, 3, \ldots, 10$*, si rappresenti graficamente tale funzione e si commentino i risultati ottenuti.*

Soluzione. Una possibile implementazione è la seguente:

```
function z=mfzero(m)

for i=1:length(m)
    if m(i)<.8 x0=pi;
    else x0=m(i);
    end
    z(i)=fzero('zfun',x0,optimset('Display','final',...
         'TolX',eps),m(i));
end plot(m,z);
```

dove *zfun* è

```
function F=zfun(x,m)

F=sin(x)-m*x;
```

Assegnato il vettore

```
>> m=[.0:.01:1,2:10]';
```

si ottiene il grafico delle radici z in funzione di m riportato in figura 7. L'istruzione di selezione del controllo *if* in *mfzero* consente di far convergere *fzero* sempre verso lo zero di *zfun* positivo più piccolo e non verso il suo simmetrico negativo, situazione che si può presentare quando $m \simeq 1$. La ragione per cui questo accade risiede nell'algoritmo su cui si basa la *fzero* di MATLAB, ovvero una combinazione dei metodi di bisezione, delle secanti e dell'interpolazione quadratica inversa (vedi sezione Istruzioni MATLAB). In particolare la ricerca dello zero parte con la valutazione di *zfun* in x_0 per poi cercare alternativamente a sinistra e a destra di x_0 un valore x_* in cui *zfun* cambi segno. Si può osservare tramite l'opzione *iter* di *Display* in *options* che tale ricerca procede su intervalli sempre più ampi (passi da 2 a 24, esempio con $x_0 = \pi$):

```
>> z=mfzero(.99)'

 Func-count       x            f(x)          Procedure
    1          3.14159      -3.11018          initial
    2          3.05273      -2.93347          search
```

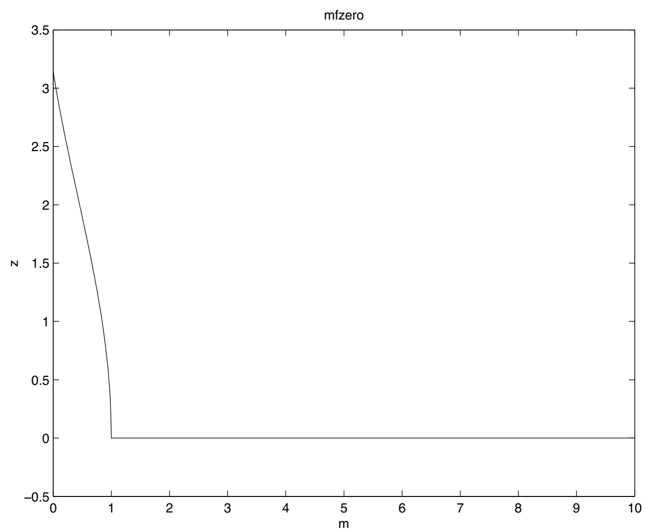

Figura 7: *mfzero* per $m = 0, 0.01, 0.02, \ldots, 0.98, 0.99, 1, 2, 3, \ldots, 10$.

```
 3         3.23045      -3.28689         search
 4         3.01593      -2.86044         search
 5         3.26726      -3.35992         search
 6         2.96388      -2.75746         search
 7         3.31931       -3.4629         search
 8         2.89027      -2.61267         search
 9         3.39292      -3.60768         search
10         2.78616      -2.41031         search
11         3.49702      -3.81005         search
12         2.63894      -2.13079         search
13         3.64425      -4.08956         search
14         2.43073      -1.75394         search
15         3.85245      -4.46642         search
16         2.13628      -1.27059         search
17          4.1469      -4.94976         search
18         1.71987     -0.713762         search
19         4.56332      -5.50659         search
20         1.13097     -0.214837         search
21         5.15221      -6.00552         search
22        0.298148   -0.00141609         search
23         5.98504      -6.21894         search
24       -0.879646      0.100336         search

Looking for a zero in the interval [-0.87965, 5.985]
```

```
   25        -0.77065     0.0663417        interpolation
   26       -0.560185     0.0232402        interpolation
   27         2.71243      -2.26919        bisection
   28       -0.527008     0.0187883        interpolation
   29       -0.388139    0.00579113        interpolation
   30         1.16214     -0.232865        bisection
   31       -0.350521    0.00362861        interpolation
   32       -0.288367    0.00109631        interpolation
   33        0.436888   -0.00939731        bisection
   34       -0.212597  -0.000528109        interpolation
   35       -0.237231  -0.000153402        interpolation
   36       -0.246077   1.52156e-005       interpolation
   37       -0.245279 -7.72576e-007        interpolation
   38       -0.245318  -3.5623e-009        interpolation
   39       -0.245318   6.91114e-015       interpolation
   40       -0.245318 -2.77556e-017        interpolation
   41       -0.245318             0        interpolation
Zero found in the interval: [-0.87965, 5.985].

z =

   -2.453178088540259e-001
```

Quando x_* viene trovato (in questo esempio $x_* = x_{24} = -0.879646$), l'algoritmo procede per interpolazione quadratica inversa intervallato da un passo di bisezione ogni volta che *zfun* cambia segno (ad esempio ai passi 26 e 27). Se l'intervallo $[x_*, x_0]$ contiene entrambi gli zeri simmetrici rispetto all'origine di *zfun*, *fzero* converge a quello negativo. Per evitare questo bisogna scegliere il valore iniziale x_0 in modo che la fase di ricerca di x_* si concluda con un intervallo che contiene solo lo zero positivo. Ad esempio partendo con $x_0 = m = 0.99$ si ottiene lo zero positivo:

```
>> z=mfzero(.99)'

 Func-count       x           f(x)         Procedure
    1            0.99     -0.144074         initial
    2        0.961999     -0.132042         search
    3           1.018     -0.156761         search
    4          0.9504     -0.127248         search
    5          1.0296     -0.162211         search
    6        0.933997     -0.120654         search
    7           1.046     -0.170115         search
    8          0.9108     -0.111698         search
    9          1.0692     -0.181692         search
   10        0.877994     -0.099755         search
   11         1.10201      -0.19887         search
   12          0.8316    -0.0842738         search
```

```
   13          1.1484     -0.224807        search
   14        0.765989    -0.0650789        search
   15         1.21401     -0.264847        search
   16          0.6732     -0.042977        search
   17          1.3068     -0.328377        search
   18        0.541977    -0.0207266        search
   19         1.43802     -0.432444        search
   20          0.3564   -0.00393327        search
   21          1.6236     -0.608758        search
   22       0.0939543   0.000801375        search

   Looking for a zero in the interval [0.093954, 1.6236]

   23       0.0959653   0.000812425        interpolation
   24        0.859783    -0.0934841        bisection
   25        0.102546   0.000845831        interpolation
   26        0.481164    -0.0135411        bisection
   27        0.124806   0.000924304        interpolation
   28        0.302985   -0.00158459        bisection
   29        0.190449   0.000755286        interpolation
   30        0.246717 -2.81382e-005        bisection
   31        0.244696  1.23509e-005        interpolation
   32        0.245313   1.0525e-007        interpolation
   33        0.245318 -6.01591e-012        interpolation
   34        0.245318  1.94289e-016        interpolation
   35        0.245318             0        interpolation
Zero found in the interval: [0.093954, 1.6236].

z =

    2.453178088540252e-001
```

Si fa osservare che la scelta euristica di x_0 implementata in *mfzero* non è l'unica possibile per ottenere sempre la convergenza verso lo zero positivo. ■

4 Domande di verifica

- E' vero che un piccolo valore di $|f(x)|$ garantisce una soluzione accurata del problema $f(x) = 0$?
- Cosa succede all'ordine di convergenza del metodo di Newton quando la radice da approssimare è multipla?
- Supponi che il metodo di bisezione parta da un intervallo di localizzazione di ampiezza uno. Quante iterazioni sono necessarie per ottenere un'approssimazione della radice con un errore assoluto dell'ordine di 10^{-8}?

- Quale condizione sulla funzione g assicura che il metodo di iterazione funzionale $x_{k+1} = g(x_k)$ sia localmente convergente?
- Definisci il concetto ordine di convergenza p.
- Supponi che al passo k l'errore e_k sia dell'ordine di 10^{-2}, descrivi l'andamento dell'errore per una successione che converge quadraticamente.

Riferimenti bibliografici

[BBCM92] Bevilacqua, R., Bini, D., Capovani, M. e Menchi, O. (1992) *Metodi Numerici*, Zanichelli, pagg.104-118, 125-134, 139-142 (senza dimostrazioni).

[Mon98] Monegato, G. (1998) *Fondamenti di Calcolo Numerico*, Clut, pagg.181-198 (senza dimostrazioni).

[NPR01] Naldi, G., Pareschi, L. e Russo, G. (2001) *Introduzione al Calcolo Scientifico*, Mc Graw Hill Ed., pagg.203-230. (senza dimostrazioni).

[Hea02] Heath, M.T. (2002) *Scientific Computing*, Mc Graw Hill Ed., pagg.216-236 (senza dimostrazioni).

[Ste96] Stewart, G.W. (1996) *Afternotes on Numerical Analysis*, SIAM Ed., pagg.1-32 (senza dimostrazioni).